ค่าคะแนนความแตกต่างมัธยฐานของหน้าต่างอนุกรมย่อยสำหรับค่าผิดปกติแบบบริบทบนอนุกรมเวลา

นายอาทิตย์ สกุลเมือง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559

MEDIAN-DIFFERENCE WINDOW SUBSERIES SCORE FOR CONTEXTUAL ANOMALY ON TIME SERIES

Mr. Artit Sagoolmuang

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Applied Mathematics and Computational Science

Department of Mathematics and Computer Science

Faculty of Science

Chulalongkorn University

Academic Year 2016

Thesis Title	MEDIAN-DIFFERENCE WINDOW SUBSERIES SCORE FOR CONTEXTUAL ANOMALY ON TIME SERIES
By	Mr. Artit Sagoolmuang
Field of Study	Applied Mathematics and Computational Science
Thesis Advisor	Assistant Professor Krung Sinapiromsaran, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

..... Dean of the Faculty of Science
(Associate Professor Polkit Sangvanich, Ph.D.)

THESIS COMMITTEE

..... Chairman
(Assistant Professor Boonyarit Intiyot, Ph.D.)

..... Thesis Advisor
(Assistant Professor Krung Sinapiromsaran, Ph.D.)

..... Examiner
(Jiraphan Suntornchost, Ph.D.)

..... External Examiner
(Assistant Professor Chumphol Bunkhumpornpat, Ph.D.)

อาทิตย์ สกุลเมือง : ค่าคะแนนความแตกต่างมัธยฐานของหน้าต่างอนุกรมย่อยสำหรับค่าผิดปกติแบบบริบทบนอนุกรมเวลา. (MEDIAN-DIFFERENCE WINDOW SUB-SERIES SCORE FOR CONTEXTUAL ANOMALY ON TIME SERIES) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร. กรุง สินอภิรมย์สราญ, 79 หน้า.

การตรวจจับค่าผิดปกติบนอนุกรมเวลาเป็นหนึ่งในหัวข้อที่น่าสนใจในการทำเหมืองข้อมูล โดยมีจุดประสงค์เพื่อค้นหาค่าข้อมูลซึ่งมีความแตกต่างจากข้อมูลส่วนใหญ่เรียกว่าค่าผิดปกติ ในวิทยานิพนธ์ฉบับนี้ คะแนนค่าผิดปกติแบบใหม่เรียกว่า ค่าคะแนนความแตกต่างมัธยฐานของ หน้าต่างอนุกรมย่อย (เอ็มดีดับเบิลยูเอส) ถูกนำเสนอกับขั้นตอนวิธี พร้อมกับพารามิเตอร์ของ ความยาวหน้าต่างแนะนำ เพื่อตรวจจับค่าผิดปกติแบบบริบทบนอนุกรมเวลา การคำนวณทำได้ โดยการลบกันของ ค่ากลาง-หน้าต่างกับค่ามัธยฐานของทุกค่าข้อมูลในหน้าต่างปัจจุบัน ขั้นตอนวิธีเอ็มดีดับเบิลยูเอสที่นำเสนอใช้การปรับมัธยฐานของหน้าต่างอนุกรมย่อย ณ ขณะนั้น เพื่อคงความซับซ้อนของเวลาเชิงเส้น สองเกณฑ์ค่าผิดปกติถูกประยุกต์มาจากกฎพิสัยระหว่างควอร์ไทล์ ผลการทดลองแสดงให้เห็นว่าเอ็มดีดับเบิลยูเอสมีประสิทธิภาพที่สุด ทั้งบนชุดข้อมูลเกณฑ์มาตรฐานสังเคราะห์และชุดข้อมูลเกณฑ์มาตรฐานโลกจริงจากยะฮู้ (Yahoo) และนูเมนต้า (Numenta) เปรียบเทียบกับวิธีตรวจจับค่าผิดปกติอื่นๆที่มีอยู่ นอกจากนั้น ขั้นตอนวิธีเอ็มดีดับเบิลยูเอสยังคงมีความเร็วกว่าขั้นตอนวิธีอื่นอย่างมากบนชุดข้อมูลขนาดใหญ่

ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	ลายมือชื่อนิสิต	.....
		ลายมือชื่อ อ.ที่ปรึกษาหลัก	.....
สาขาวิชา	คณิตศาสตร์ประยุกต์และวิทยาการคณนา		
ปีการศึกษา	2559		

5872096223 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE, KEYWORDS : CONTEXTUAL ANOMALY ON TIME SERIES / ANOMALY SCORE / MEDIAN-DIFFERENCE / ANOMALY DETECTION

ARTIT SAGOOLMUANG : MEDIAN-DIFFERENCE WINDOW SUBSERIES SCORE FOR CONTEXTUAL ANOMALY ON TIME SERIES. ADVISOR : ASSISTANT PROFESSOR KRUNG SINAPIROMSARAN, Ph.D., 79 pp.

Anomaly detection on time series is one of the exciting topics in data mining. The aim is to find a data point which is different from the majority, called an anomaly. In this thesis, a novel anomaly score called Median-Difference Window subseries Score (MDWS) is proposed with its algorithm together with the parameter of the recommended window length for detecting the contextual anomalies on time series data. It is computed as the subtraction of the middle-window point with the median of all data points within the current window. The proposed MDWS algorithm is implemented as the median-update of the current window subseries to maintain the linear time complexity. Two anomaly thresholds are applied from interquartile range rule. The experimental results show that the MDWS has the highest performance on both synthetic and real world benchmark datasets from Yahoo! and Numenta comparing with others existing anomaly detection methods. Moreover, MDWS algorithm is also faster than other algorithm on the large dataset.

Department	: Mathematics and Computer Science	Student's Signature	.....
		Advisor's Signature	.....
Field of Study	: Applied Mathematics and Computational Science		
Academic Year	: 2016		

ACKNOWLEDGEMENTS

Firstly, I am very thankful to my thesis advisor Assistant Professor Dr. Krung Sinapiromsaran for his salutary suggestions and encouragement from the start to finish of this thesis. He did not just provide guidance for this research, but he also gave various valuable advice in my life. I could not complete the Master degree program without his support.

Next, I would to thank my thesis committees, Assistant Professor Dr. Boonyarit Intiyot, Dr. Jiraphan Suntornchost and Assistant Professor Dr. Chumphol Bunkhumpornpat for their useful comments and suggestions on my thesis.

Moreover, I want to thank Applied Mathematics and Computational Science Program in the Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University and Science Achievement Scholarship of Thailand (SAST) for financial and technical support.

Finally, I am thankful to my family and my friends, especially Senee Kitimoon, Warunya Kiang-ia, Panote Songwattanasiri and everybody in the AMCS laboratory for all support throughout the period of this thesis.

CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER I

INTRODUCTION

This chapter explains the problem description, the motivation, and the scope of this research including its related work.

1.1 Motivation and Literature Surveys

Data mining or knowledge discovery in databases (KDD) is a remarkable field of computer science and information technology. The goal is to find or extract or "mine" some hidden knowledge from the large amounts of data. Normally, the data mining task can be divided into two categories: description and prediction [14]. A descriptive task attempts to explain the nature and the properties of the dataset such as an association rule and clustering. For a predictive task, the model is built from a historical data for predicting unknown data points such as a classification task and a linear regression.

Time series domains, such as daily closing stock prices, aviation demand and online website click rate, concern with finding the best forecasting model to predict value in the future. There are many techniques to build a forecasting model, e.g. ARIMA [2], Exponential Smoothing [4], and GARCH [1]. Occasionally, the time series data contains some data points having different characteristics with respect to their context, which may cause the low forecasting accuracy of the model. Those deviated data points are called contextual anomalies.

Note the exponential smoothing model with the smoothing factor $0 < \alpha < 1$ is defined as

$$f_t \ = \ \alpha y_t \ + \ (1-\alpha) f_{t-1}, \text{ for } t > 1$$

where y_t is the original time series value and f_t is the forecasting value at time t, and

$f_1 = y_1$. If the time series has anomaly at time t_1, then the forecasting value f_{t_1} will be distorted, causing the next forecasting values f_{t_1+1}, $f_{t_1+2}, ...$ to be distorted as well. Hence, it is necessary to identify this anomaly and modify it appropriately before building the forecasting model.

Furthermore, the anomaly detection on time series is also useful in many applications, for example, detecting anomalous heartbeat pulses using electrocardiogram (ECG) data [21, 8], attack detection in network systems [10, 30] and credit card fraud detection [11].

Anomaly detection is one of the challenging problems in data mining. There are many techniques to detect anomalies on a non-temporal domain, e.g. statistical based [15, 12], distance based(DB) [19, 25], and density based [3, 16]. For the time series domain, the anomaly researches have received little attention. Most of them require a training dataset with the target class and build a detection model based on various concepts such as similarity based [24, 26] and prediction based [22, 23, 33], see [5, 7]. Some situations such as the anomalies in the customer transactions have no well-defined target, the anomaly detection technique without the target is needed.

In 1975, Rosner [27] proposed the extreme studentized deviate (ESD) many-outlier for detecting from 1 to k anomalies on a univariate dataset. The conclusion from the ESD many-outlier shows the reliability of their method on a dataset without anomaly. Then in 1983, Rosner [28] improved his procedure and suggested the generalized ESD many-outlier for a dataset with some anomalies. Since this procedure was established for the non-temporal dataset, it was not appropriate to use for detecting anomalies on the time series data. In 2014, Vallis et al. [29] applied the generalized ESD many-outlier on the time series data. Their method split the dataset into non-overlapping window, then it performed the generalized ESD many-outlier on the residual after removing trend and seasonality of each window. In 2015, Kejariwal [17] provided the open-source R package for detecting anomalies on the time series called Seasonal Hybrid ESD (S-H-ESD) based on the method of Vallis.

Nowadays, the anomaly detection methods often assign an anomaly score to each data point instead of directly assign a label (normal or anomaly). Therefore, the interesting anomaly score on the time series data called the Furthest Neighbor Window Subseries (FNWS) was proposed by Kitimoon et al. [18] in 2016. It used sliding window subseries of length w and represented them using three quartiles subtracting with the first data point of the window. The score was computed using the furthest k-nearest neighbor distance. Additionally, the threshold is set to the upper quartile plus the triple of interquartile range for detecting anomalies.

In this thesis, we propose a new anomaly score for detecting contextual anomaly on time series data, called the MDWS (Median-Difference Window subseries Score). It relies on the idea that the anomalies should have different values from their normal surrounding context, both preceding data points and the succeeding data points. The MDWS requires the concept of window subseries and the concept of representative of the data distribution. The window subseries covers data points preceding and succeeding of the examined data point along the time dimension within a specified window size which is governed by its representative.

MDWS distinguishes between the normal points which are similar to the representative and the anomalies which are different. Since the median is not influenced by anomalies and missing values [13], it is selected as the representative of the window subseries. The MDWS is computed using the subtraction of the examined data points with the representative of its window. To generate a score, the MDWS algorithm uses a median update concept to minimize a time complexity. Finally, an interquartile rule for specifying the anomalies from the dataset is applied.

1.2 Research Objectives

The goal of this research is to construct the novel anomaly score, called the Median-Difference Window subseries Score (MDWS). It is designed to detect point anomalies and contextual anomalies on the time series data with trend, seasonality and noises. In addition, the MDWS algorithm is proposed for computing the MDWS of a data point in the dataset as the unsupervised technique. Finally, the thresholds are given for specifying the anomalies. The MDWS algorithm is implemented, and experimented on the real world dataset and synthetic datasets, then it compares with other methods using the precision, the recall, and the F1-measure as their performance measures.

1.3 Thesis Overview

The remainder of this thesis is organized as follows. In chapter II, some necessary background knowledge is explained. For chapter III, the formal definition of the MDWS together with its algorithm and thresholds are proposed. Next, the experimental results show the accuracy performance of the MDWS and the time efficiency of the MDWS algorithm in chapter IV. Last chapter provides the discussion and conclusion of this work.

CHAPTER II

BACKGROUND KNOWLEDGE

In this chapter, the preliminaries of this thesis are described which are split into four parts, basic statistical knowledge, a formal definition of the time series and its related properties, the anomaly detection concept, and two other effective methods for detecting anomalies on the time series data.

2.1 Statistics

Statistics is an important background basis for data mining tasks. In this thesis, the statistical knowledge for describing the distribution of the data points and separating the anomalies out of the normal data points are described in this section.

2.1.1 Measures of Position

Univariate statistics can specify the position of each data point in a given dataset using the measures of position. Most of them sort the dataset from the lowest value to the highest value and divide them into equal groups, then specify each position by the point that separate each group. The various measures of position are presented as follows:

Quartiles

Quartiles of the numeric dataset are three points that divide the dataset into four groups of equal sizes. The first quartile is denoted by Q_1 which covers 25% of the data points less than it and 75% of the data points greater than it. The second quartile is denoted by Q_2 which covers 50% of the the data points less than it and 50% of the data points greater than it. The third quartile is denoted by Q_3 which covers 75% of the data points less than it and 25% of the data points greater than it.

Example 2.1**.** Let $X = \{0,\ 1,\ 4,\ 5,\ 7,\ 10,\ 100\}$ be the sorted univariated dataset of 7 data points then,

- the first quartile : $Q_1 = 1$
- the second quartile : $Q_2 = 5$
- the third quartile : $Q_3 = 10$

Boxplot is a graphically representation of a dataset using quartiles which is shown in Fig. 2.1.

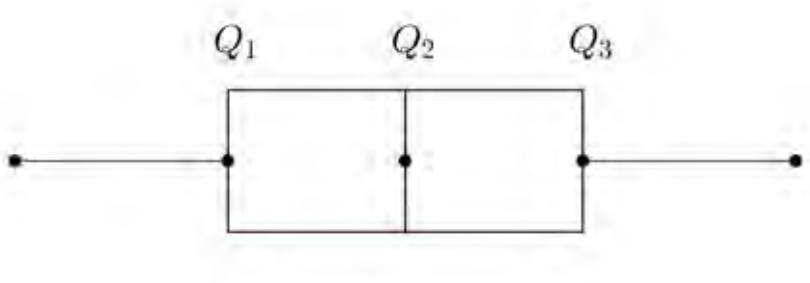


Figure 2.1**:** Box Plot

Deciles

Deciles of the sorted dataset are the nine points that split the dataset into ten equal-size groups. The first decile is denoted by D_1 which covers 10% of the data points less than it and 90% of the data points greater than it. The position of $D_2, ..., D_9$ are defined similarly.

Percentiles

For the percentiles, they divide the dataset into hundred groups which consist of 1% of the data points. The position of each percentile P_1, P_2, ..., P_{99} are defined similar to quartiles and deciles, such as the first percentile P_1, there are 1% of the data points less than it and 99% of the data points greater than it.

The following statements show some relationship between quartiles, deciles and per-

centiles.

$$
\begin{aligned}
Q_1 &= P_{25}, \\
Q_2 &= D_5 = P_{50}, \\
Q_3 &= P_{75}
\end{aligned}
$$

Standard Scores

There are some measures of position without dividing the dataset into groups such as the standard score or z-score. It identifies the position of each data point by the mean (μ) and the standard deviation (σ). The standard score of the given data point x is calculated by the following equation:

$$z = \frac{x - \mu}{\sigma}$$

For the dataset which contains some anomalies, the standard score is not appropriate for specifying the position of a data point because the effect of the anomalies makes the mean deviate from the majority which is shown in the next section. This thesis uses another measure of position, which are quartiles.

2.1.2 Measures of Central Tendency

To describe the nature of the univariate dataset, the measure of central tendency is used. This thesis uses the measures of central tendency for generating the representative of a window subseries. The common measures of central tendency are mean, median, and mode which are defined as follows:

Mean

The mean (or arithmetic mean or average) is the most commonly used and readily understood measure of central tendency. The mean of a finite dataset is defined as the

sum of all data points and divided it by the total number of them. Symbolically, let x_1, x_2, ..., x_n be the sequence of n data points then the mean is represented by $\bar{x}$ which is defined as:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

Median

The median is the value which separates the higher half of a dataset from the lower half. In other words, the median is defined as the middle value in the list of sorted data points. Mathematically, let $a_1, a_2, ..., a_n$ be the sorted sequences of n data points, the median is symbolized by $\tilde{a}$ which is defined as:

$$\widetilde{a} = \begin{cases} a_{\dfrac{n+1}{2}}, & \text{if } n \text{ is odd} \\ \\ \dfrac{a_{\frac{n}{2}} + a_{\frac{n}{2}+1}}{2}, & \text{if } n \text{ is even.} \end{cases}.$$

Note that, the median is equal to the second quartile (Q_2), the fifth decile (D_5) and the fiftieth percentile (P_{50}), i.e.

$$\widetilde{X} \;=\; Q_2 \;=\; D_5 \;=\; P_{50}.$$

Mode

The mode is the value that appears most often in a sequence of observations. It is not necessarily unique to a given dataset. Moreover, the mode is the only measure of central tendency that can be used for qualitative variable, but it is not effective for the continuous dataset.

Example 2.2**.** Let $-2,\ -1,\ 0,\ 1,\ 2,\ 3,\ 4,\ 4,\ 4,\ 5$ be the sequences of 10 data points, then

- mean $= \dfrac{-2-1+0+1+2+3+4+4+4+5}{10} = \dfrac{20}{10} = 2$
- median $= \dfrac{2+3}{2} = \dfrac{5}{2} = 2.5$
- mode $= 4$ (repeated three times)

Example 2.3**.** Let $0, 0, 1, 1, 1, 2, 2, 2, 2, 3, 976$ be the sequences of 10 normal data points and 1 anomaly, then

- mean $= \dfrac{0+0+1+1+1+2+2+2+2+3+976}{11} = \dfrac{990}{11} = 90$
- median $= 2$
- mode $= 2$ (repeated four times)

Example 2.4**.** Consider the sequences of 30 normal data points and 1 anomaly which are presented by the circle points in the figure below. The mean and median of this dataset are shown by the lines as follows:

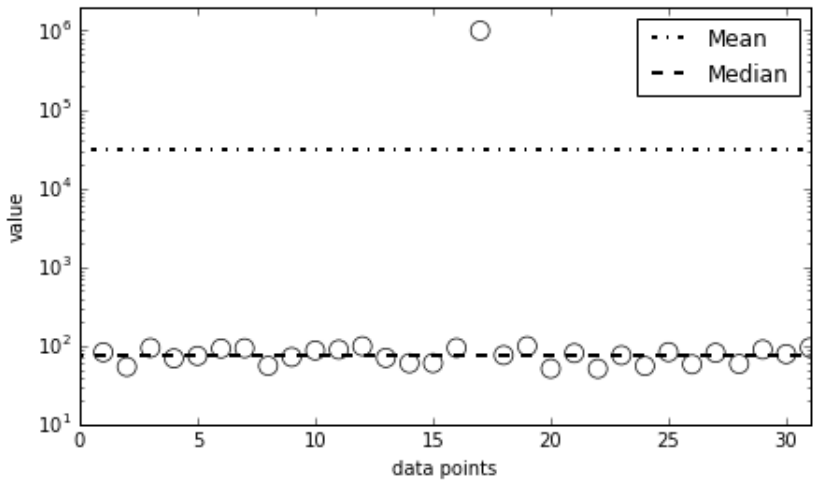


Figure 2.2**:** The mean and median of the dataset which contains an anomaly.

Example 2.5**.** Demonstrate the effect of varying a single point from the rest of cluster data points. Let $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, X$ be the sequences of 11 data points. The mean and median of this dataset which varies by X from 10 to 100 are shown by the figure below:

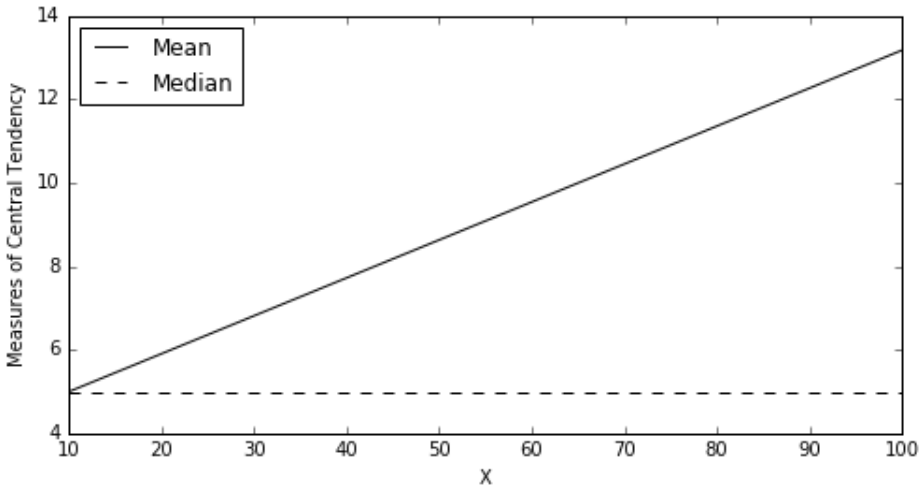


Figure 2.3**:** The mean and median of this dataset which varies by X.

From Examples 2.3 - 2.5, the median is more robust than the mean for a dataset. Note that, the median can tolerate up to 50% of anomalies in the dataset. [13]. Consequently, MDWS used the median for representing the distribution of each window subseries.

2.1.3 **Measures of Dispersion**

Since the measures of central tendency use only a single value for describing a dataset, in some situations, two datasets having the same mean or median may have different spread. To characterize the spread of the dataset, the measures of dispersion are used. This section will show four common measures of dispersion, i.e. range, interquartile range, standard deviation and median absolute deviation.

Range

The range is the easiest and most rough measure of dispersion, it is computed by the difference between the highest and lowest values within the dataset.

Interquartile Range

The interquartile range (IQR) is the middle fifty or midspread of the dataset.

Statistically, the interquartile range is defined by the difference between the third quartile (Q_3) and the first quartile (Q_1).

$$IQR \ = \ Q_3 \ - \ Q_1$$

Note that, the interquartile range is used as the length of the box from the box plot as illustrated in Figure 2.4.

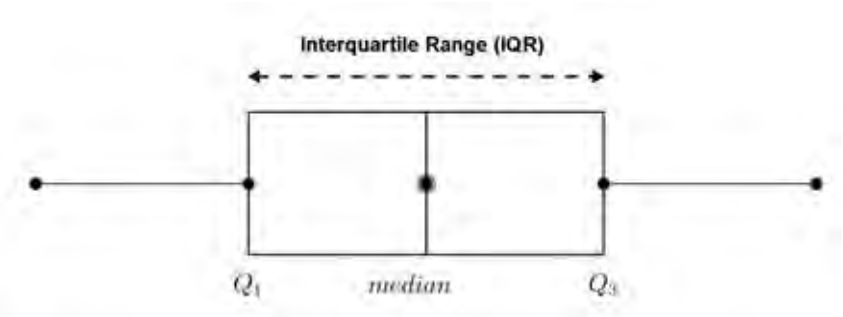


Figure 2.4**:** Interquartile Range

Standard Deviation

The standard deviation (s) is the most common measure of dispersion that is used to quantify the amount of variation or dispersion of a dataset. It is computed as the root of the bias mean square of the difference between each data point and its mean. Mathematically, let x_1, x_2, ..., x_n be the sequences of n data points with the mean $\bar{x}$, the standard deviation is defined as

$$s \ = \ \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Note that, the square of standard deviation (s^2) is called the variance.

Median Absolute Deviation

The median absolute deviation (MAD) is a robust measure for determining the spread of a dataset. It calculates as the median of the difference between each data point

and the median of the dataset. For the dataset, the median absolute deviation is defined as follows:

$$MAD \ = \ \text{median}\{|x_i - \tilde{x}||\text{for } i = 1, ..., n\}$$

The next example shows that the effect of the anomalies on each measure of dispersion.

Example 2.6**.** Let $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, X$ be the sequences of 11 data points. The range, interquartile range, standard deviation and median absolute deviation of the dataset which varies by X from 10 to 100 are shown by the figure below:

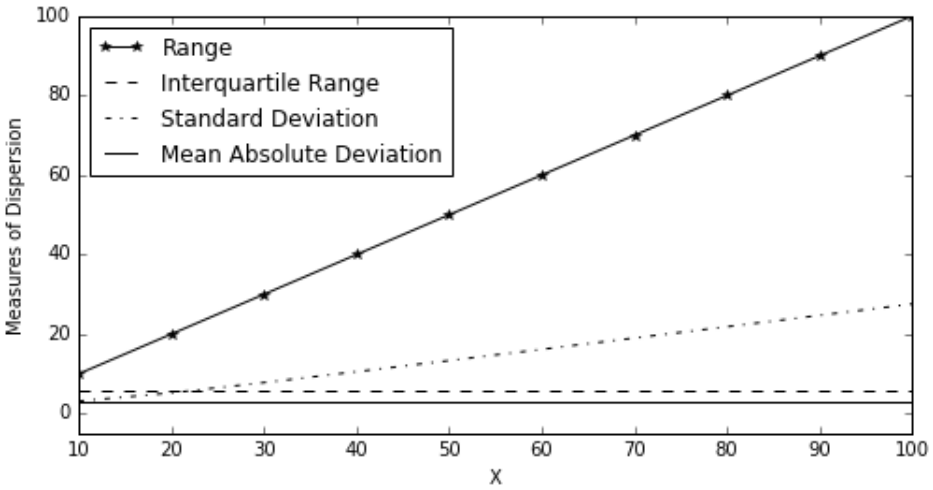


Figure 2.5**:** The range, interquartile range, standard deviation and median absolute deviation of the dataset which varies by X.

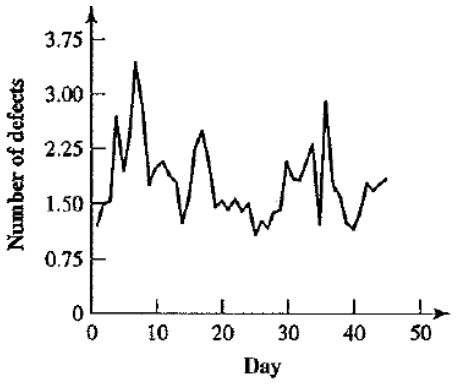
Example 2.6 shows that the range is sensitive to the value of X, similar to the standard deviation. On the other hand, the interquartile range and the median absolute deviation remain constant throughout the value changes of X.

2.2 Time Series

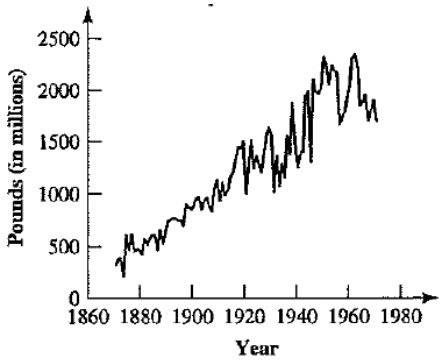
"A time series is an ordered sequence of observations. Although the ordering is usually through time, particularly in terms of some equally spaced time intervals," [31].

Time series is an important class of data objects which appears in a variety of fields

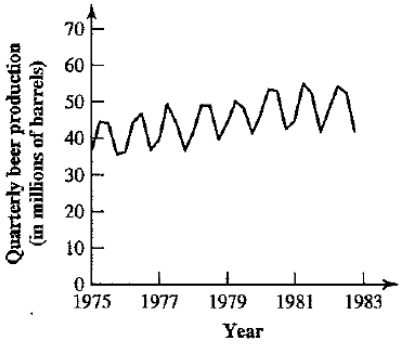
(Figure 2.6). In the business field, for example, the time series data is used for presenting a daily closing stock prices, credit card transaction, quantity of product and ATM daily cash. In the communications field, it appears in the aviation demand, online website click rate and the volume of tweets. Furthermore, an electrocardiogram (ECG), crime rate, wind speed, gasoline demand, organism popualtion and others are presented as time series. The increasing of time series data has initiated various challenging problem and explication. In this section, the formal definition and its related properties are presented.



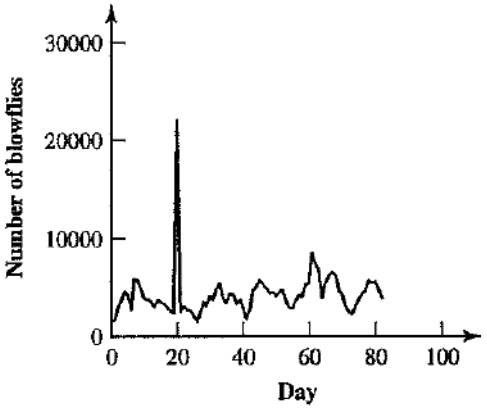
(a) Number of truck manufacturing defects



(b) U.S. tobacco production



(c) U.S. beer production



(d) Contaminated blowfly data

Figure 2.6**:** The applications of Time Series [31].

Definition 2.1. *A restricted time series* $Y_t = \{y_0, y_1, ..., y_{n-1}\}$ is an ordered set of n observed values with respect to their time stamp $t = \{t_0, ..., t_{n-1}\}$. For short it can be written only as $Y = \{y_0, y_1, ..., y_{n-1}\}$.

Example 2.7. Let Y = {-9, 4, 2, 3, -5, 3, 1, -6, 0, 2, 9, -2, 4, 4, 8, -1, 7, -6, 1, 3} be the time series of range 20 with respect to $t = \{10, 20, ..., 200\}$. It is shown in Figure 2.7.

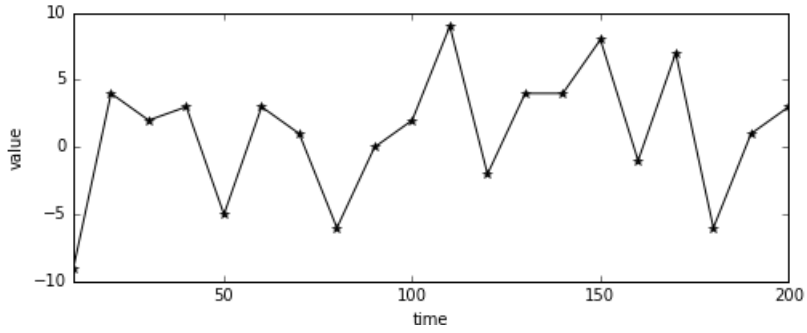


Figure 2.7**:** The graph of time Series in Example 2.7 .

2.2.1 **The Components of Time Series**

In a time series analysis, the time series is usually decomposed into four components, i.e. trend (T), cycle (C), seasonality (S) and irregularity (I). A model may express some or all components which are combined in different ways, e.g. an additive model and multiplicative model :

$$Y_t = T_t + C_t + S_t + I_t$$
$$Y_t = T_t \times C_t \times S_t \times I_t$$

The additive model is more appropriate than the multiplicative model when the magnitude of the seasonality or the variation around the trend and cycle does not vary with the level of the time series. On the other hand, if the variation in the seasonality or the variation around the trend and cycle appears to be proportional with the level of the time series such as in an economics, then a multiplicative model is more appropriate.

Trend

Trend (T) variation is the main component of the time series which is referred as the long-term increasing or decreasing movement in the time series data. It may not necessary be linear, and it may be either exponential or damped or mixed. For example, the increase in aviation demand each year and the decrease in deaths due to advances in science.

Cycle

The cyclical component (C) shows an up and down oscillation around a given trend that is not of the fixed period. The cycle is mostly observed in economic data, the duration of it depends on a type of business or industry.

Seasonality

Seasonality (S) is the component of a time series that represents the variations of periodicity, it exists when the dataset is influenced by a seasonal factor. The seasonality always has a fixed and known period, e.g. a quarter of a year, a month, or biweekly. For example, the export volume of agricultural products in a month of the year and the traffic on roads at different times of the day.

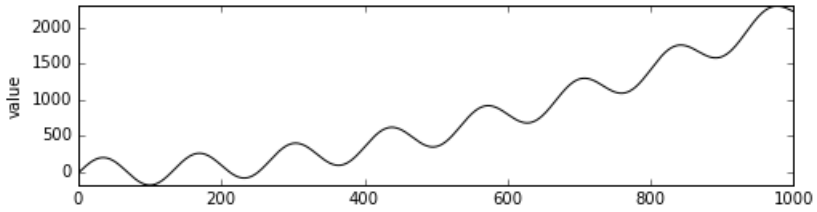
Irregularity

Ideally, any time series data has the apparent structure with the trend, the cycle and the seasonality. But in fact, there are some unwanted components that make it deviate from the ideal, called irregularity (I). The irregular component is known as noise, it is unpredictable and uncontrollable which cannot be explained by other components. In stochastic process, it is normally referred as the white noise. The white noise process is the independent and identically distributed (i.i.d.) random vector with zero mean ($\mu = 0$) and constant variance σ^2. In particular, if the white noise process has a normal distribution (i.e. $N(0, \sigma^2)$), this process is known as the Gaussian white noise.

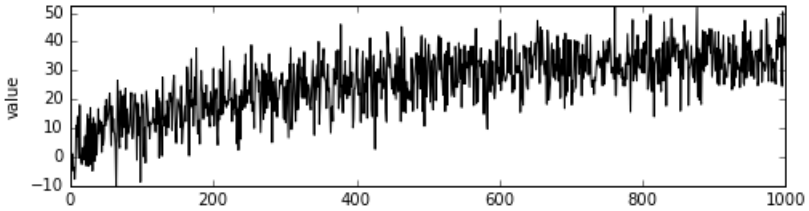
Time series with the trend, seasonality and irregularity are shown in Figure 2.8.

2.2.2 Window Subseries

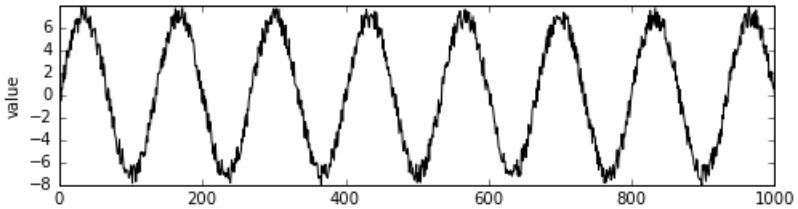
The time series data has been influenced by time which can be used to identify the context surrounding the considered point. The concept of window subseries is adapted to limit the group of adjacent data points along the time dimension. In this section, two types of the window subseries are defined, i.e. non-overlapping window subseries and



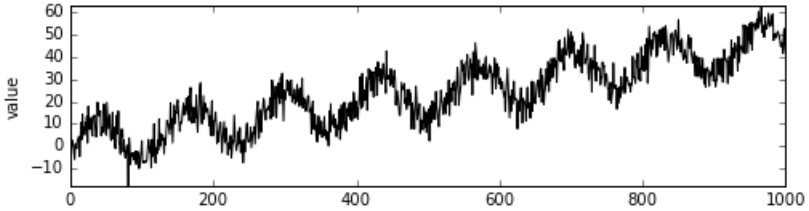
(a) Time series with exponential trend and sesonality.



(b) Time series with logarithmic trend and Guassian white noise.



(c) Time series with sesonality and uniform white noise.



(d) Time series with linear trend, sesonality and Guassian white noise.

Figure 2.8**:** Four examples of time series with mixture of various components.

sliding window subseries. Furthermore, the sliding window is divided with two indicators, the first value and the middle value.

Non-overlapping Window Subseries

First, the definition of non-overlapping window subseries is defined. It divides the time series data into equal disjoint groups. Subset of data points are contained in a

window.

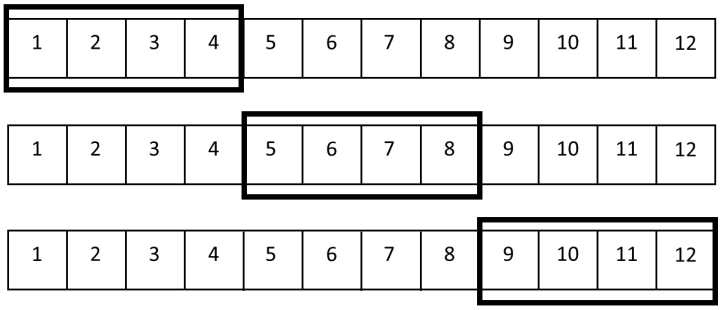


Figure 2.9**:** The non-overlapping window subseries.

Definition 2.2**.** Given a time series $Y = \{y_0, y_1, ..., y_{n-1}\}$ of length n, *a non-overlapping window subseries* W_i^n of Y for $i = 0, 1, ..., \lfloor \frac{n}{w} \rfloor - 1$ is an ordered subset with length $w \leq n$ of Y defined by $W_i^n = \{y_{i \cdot w}, y_{i \cdot w+1}, ..., y_{(i+1) \cdot w-1}\}$.

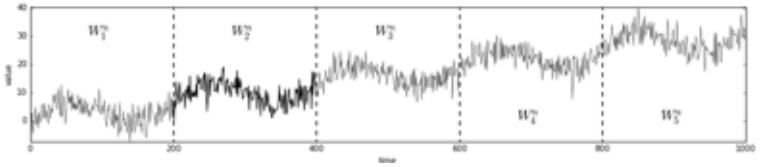


Figure 2.10**:** The example of non-overlapping window subseries with length 200 of the time series Y.

Sliding Window Subseries

The sliding window is more powerful, robust and flexible than the non-overlapping window. In this work, the step length for sliding is set to be one. It means that when the window is shifted, an oldest data point is removed from the window and a new data point is introduced into the window. Hence a typical data point appears in w windows.

The sliding window can be indexed by the first value or indexed by the middle value.

Definition 2.3**.** Given a time series $Y = \{y_0, y_1, ..., y_{n-1}\}$ of length n, *a sliding window subseries (indexed by the first value)* W_i^f of Y for $i = 0, 1, ..., n - w$ is an ordered subset with length $w \leq n$ of Y defined by $W_i^f = \{y_i, y_{i+1}, ..., y_{i+w-1}\}$.

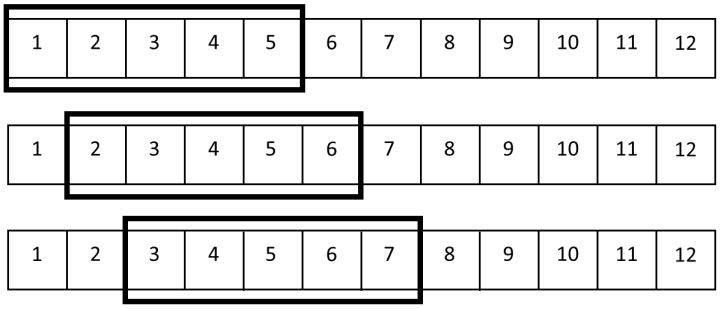


Figure 2.11**:** The sliding window subseries.

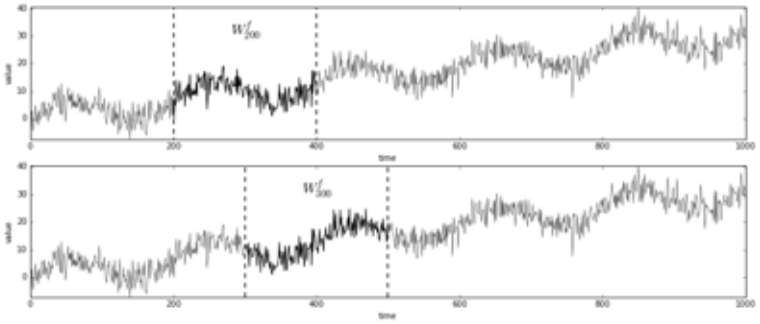


Figure 2.12**:** Examples of sliding window subseries (indexed by the first value) with length 200 of the time series Y.

Note that, there is no index at the end of the sliding window with indexed by the first value, because it does not have enough succeeding data points.

Next, the sliding window subseries indexed by the middle value is proposed as following definition.

Definition 2.4. Given a time series $Y = \{y_0, y_1, ..., y_{n-1}\}$ of length n, *a sliding window subseries (indexed by the middle value)* W_i of Y for $i = k, k+1, ..., n-k-1$ is an ordered subset with length $2k+1$ of Y when $k \leq \frac{n-1}{2}$, defined as $W_i = \{y_{i-k}, ..., y_i, ..., y_{i+k}\}$. The middle value y_i of window subseries W_i is called middle-window point.

Note that, the index of W_i can not be less than k or greater than $n-k-1$, because it does not have enough surrounding data points.

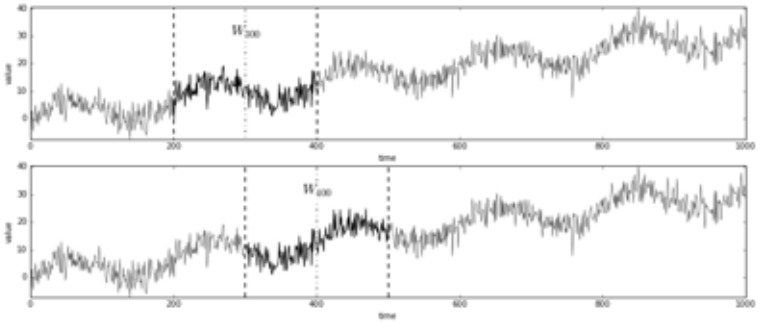


Figure 2.13**:** Examples of sliding window subseries (indexed by the middle value) with $k = 100$ of the time series Y.

2.3 Anomaly Detection

Anomaly detection is one of exciting topics in data mining, machine learning and artificial intelligence. The aim is to find the data point which deviates from the majority, called an anomaly or outlier. The most cited definition of anomaly is from the Hawkins's definition [15] which is stated that "Outlier [or "anomaly"] is observation that deviates so much from other observations". This section will discuss two issues about anomaly detection, i.e. the type of anomaly and the type of techniques that used for detecting anomalies.

2.3.1 Type of Anomaly

Since varieties of methods detect different characteristics of anomaly, then they need to clearly define the type of anomaly first. In 2009, Chandola et al. [6] divided an anomaly into three categories.

Point Anomaly

First, a point anomaly is a data point which is very different from the rest of the dataset. It is the simplest type of anomaly which appears in many research. For example, the point anomalies are found in credit card fraud and breast cancer cells.

In Figure 2.14, the point o_1 is far from the normal region N_1 and N_2 then it is

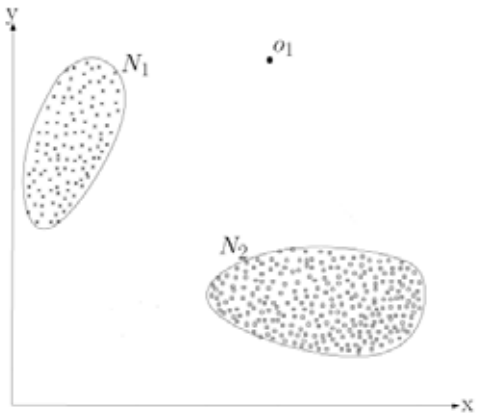


Figure 2.14**:** The point anomaly on the two dimensional dataset [6].

identified as a point anomaly.

Contextual Anomaly

Contextual anomaly is a data point which deviates respect to its context. It mostly founds in the time series data, such as the network intrusion and unusual transaction.

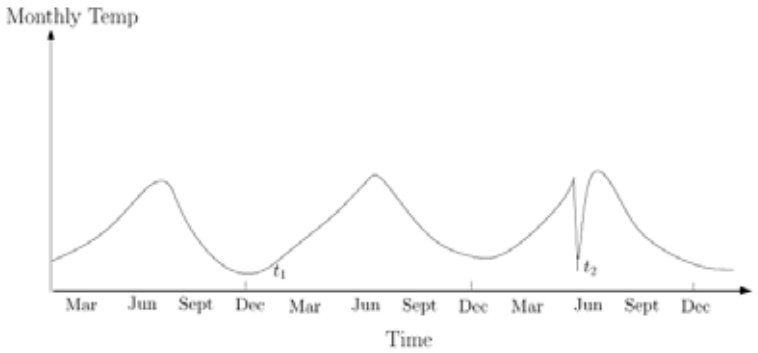


Figure 2.15**:** The contextual anomaly on the temperature time series [6].

Figure 2.15 shows the identical value of t_1 and t_2. However, t_2 is very different with its context both preceding and succeeding, it is identified as the contextual anomaly.

Collective Anomaly

The collection of data points which exhibits the different characteristic with the rest of the entire dataset is called collective anomaly. The collective anomaly has been studied in the time series data, sequential data and spatial data, e.g. the electrocardiogram

(ECG).

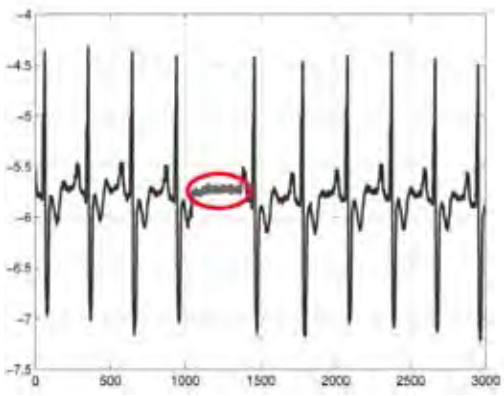


Figure 2.16**:** The collective anomaly on the electrocardiogram [6].

The circle region of Figure 2.16 denotes the collective anomaly because the same low value appears for an abnormally long time.

This thesis interests to assign the anomaly score to each data point on the time series domain for detecting the point anomalies and contextual anomalies such as Figure 2.17. Since all point anomalies are contextual anomalies, it is suffice to specify the contextual anomalies on time series.

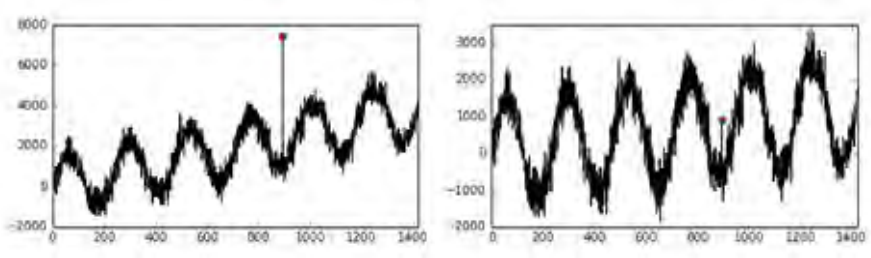


Figure 2.17**:** The point anomaly and contextual anomaly on the time series data.

2.3.2 Type of Anomaly Detection Techniques

Many techniques are proposed for detecting the anomaly which rely on various assumptions. They are also divided into three categories that are presented in this section.

Supervised Anomaly Detection

The supervised technique requires the training dataset with the target class of

normal data points and anomalies. The model is built to decide that each testing data point should be normal or anomaly. Notice that the supervised anomaly detection is recognized as the classification technique.

Semi-supervised Anomaly Detection

In some cases, the training dataset contains only the normal data points. The semi-supervised technique is proposed to build the anomaly detection model. The model works by examining the similarities of the training dataset. If the test data point exhibits a similar characteristic with the data points in the training dataset, then it is specified as the normal. On the other hand, if it is very different from the data points in the training dataset, then it is identified as the anomaly.

Unsupervised Anomaly Detection

In many situations, the anomalies are needed to be detected without the target attribute. The unsupervised technique is designed for detecting the anomaly in this situation. The main idea is to specify the data points which deviate from the majority as anomalies.

Most of anomaly detections on the time series data use the semi-supervised techniques. However, in some occasions such as the anomalies in customer transaction, the training dataset with the target attribute is not known. Then, the unsupervised technique is required, which is the aim of this thesis.

2.4 Anomaly Detection on Time Series

In this branch, two effective approaches that handle the anomalies on the time series data, i.e. Seasonal Hybrid ESD and Furthest Neighbor Window Subserires are reviewed.

2.4.1 Seasonal Hybrid ESD

Seasonal Hybrid ESD (S-H-ESD) is an open-source R package for detecting the

anomalies on the time series data which is announced by Kajariwal [17]. The S-H-ESD is built of the piecewise median anomaly detection [29] which has been adapted from the anomaly detection method on a non-temporal dataset called a generalized ESD many-outlier [28].

The Generalized ESD

The generalized ESD many-outlier performs one tail statistical hypothesis testing under the null hypothesis of no anomalies and the alternative hypotheses of $1, 2, ..., k-1$ anomalies.

$$H_0 \ : \text{ no anomalies}$$
$$H_l \ : \text{ there are } l \text{ anomalies, where } l = 1, 2, ..., k-1$$

The generalized ESD many-outlier of univariated dataset $X = \{x_1, ..., x_n\}$ is based on the statistics $R_1, ..., R_k$, which are the Extreme Studentized Deviates (ESD), where $k \leq n$ is the maximum number of anomalies. They are computed from the reduced dataset of size $n, n-1, ..., n-k+1$, respectively, i.e. for the complete dataset:

$$R_1 = \frac{max_i(|x_i - \bar{x}|)}{s},$$

where $\bar{x}$ is the average of X and s is the standard deviation of X.

For R_2, it is computed similarly from the reduced dataset of size $n-1$ obtained from removing the data point which corresponding to $max(|x_i - \bar{x}|)$ from the complete dataset. For $R_3, ..., R_k$, they are computed similarly as R_1 and R_2.

The critical values $\lambda_1, \lambda_2, ..., \lambda_k$ of the test are determined by the significant level α and computed from:

$$Pr\left\{\bigcup_{i=l+1}^{k}(R_i > \lambda_i | H_l)\right\} = \alpha$$

for $l = 0, 1, ..., k-1$.

The generalized ESD many-outlier procedure has the following form:

- If $R_i \leq \lambda_i$ for all $i = 1, 2, ..., k$, then declare that there are no anomalies.
- If $R_i > \lambda_i$ for some $i = 1, 2, ..., k$, then define $l = max\{i|R_i > \lambda_i\}$ and declare $x^{(0)}, x^{(1)}, ..., x^{(l-1)}$ to be anomalies where $x^{(0)}, x^{(1)}, ..., x^{(l-1)}$ correspond to the most extreme data points (i.e. the data points corresponding to $max(|x_i - \bar{x}|)$) in the successively reduced dataset.

Next, the method which is developed from the generalized ESD many-outlier for using in the time series data is presented.

The Piecewise Median Anomaly Detection

The process of piecewise median anomaly detection is presented in Fiqure 2.18. It splits the time series Y into non-overlaping window W^n, then remove the seasonality, and trend out of all window using STL [9] and median respectively. Next, it runs the generalized ESD many-outlier to the residual of each window. Note that, it uses the median and median absolute deviation (MAD) instead of the mean and standard deviation for computing the test statistic R_i. Then, the anomalies of each window are obtained and announced them as the anomalies.

2.4.2 Furthest Neighbor Window Subserires

Furthest Neighbor Window Subserires (FNWS) is presented by Kitimoon et al. [18] in 2016, it assigns the anomaly score for each data point in the time series data. The FNWS relies on the idea that the normal data points having the same distribution with other normal data points. To determine the distribution of each data point, the three dimensional vector computed as the difference between it and three quartiles in the specific window. Then, the anomaly score is computed with the k-nearest distance of each vector. The process of FNWS and its related definitions are introduced as follows.

The Representative Vector

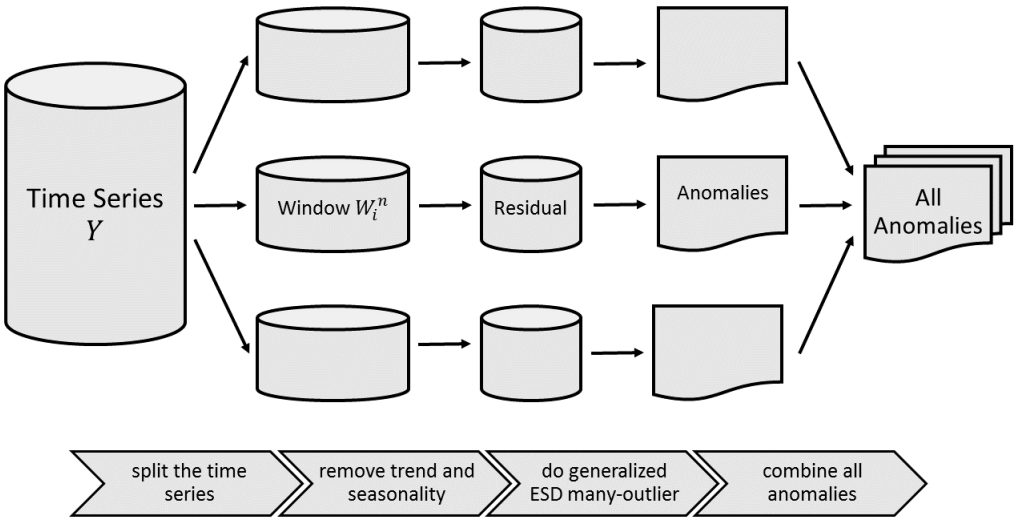


Figure 2.18**:** The process of piecewise median anomaly detection.

Let $Y = \{y_0, y_1..., y_{n-1}\}$ be the restricted time series, a representative vector rv_i of y_i is the vector of the lower quartile (Q_1), the median (Q_2), and the upper quartile (Q_3) of the sliding window subseries (indexed by the first value) W_i^s subtracting the first value y_i, i.e.,

$$rv_i = (Q_1 - y_i, Q_2 - y_i, Q_3 - y_i).$$

k-nearest distance

Given a parameter k, the k-nearest neighbor of a vector v in the dataset D is denoted by knn_v, which is consistent with the following conditions:

1. There are at least k vectors $u' \in D \setminus \{v\}$ such that $d(v, u') \leq d(v, u)$.
2. There are at most $k - 1$ vectors $u' \in D \setminus \{v\}$ such that $d(v, u') < d(v, u)$.

where d is the Euclidean distance between two n-dimensional vectors $R = (r_1, ..., r_n)$ and $S = (s_1, ..., s_n)$ defined as:

$$d(R, S) = \sqrt{\sum_{i=1}^{n} (r_i - s_i)^2}.$$

Note that, the distance between v and its k-nearest neighbors, i.e. $d(v, knn_v)$ is

called the k-nearest distance of v.

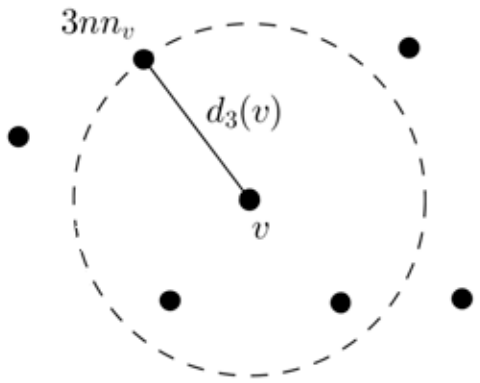


Figure 2.19**:** The 3-nearest distance of two dimensional data point v.

To detect the anomalies on the time series $Y = \{y_0, y_1, ..., y_{n-1}\}$, the FNWS works as follows:

STEP 1. For each data point y_i, the representative vector rv_i is computed from the window subseries W_i^f.

STEP 2. Calculate the anomaly score to all data points using the k-nearest distance of their representative vectors.

STEP 3. Specify the data points which have the anomaly scores greater than the suggested threshold, the upper quartile plus triple of interquartile range, as anomalies.

CHAPTER III

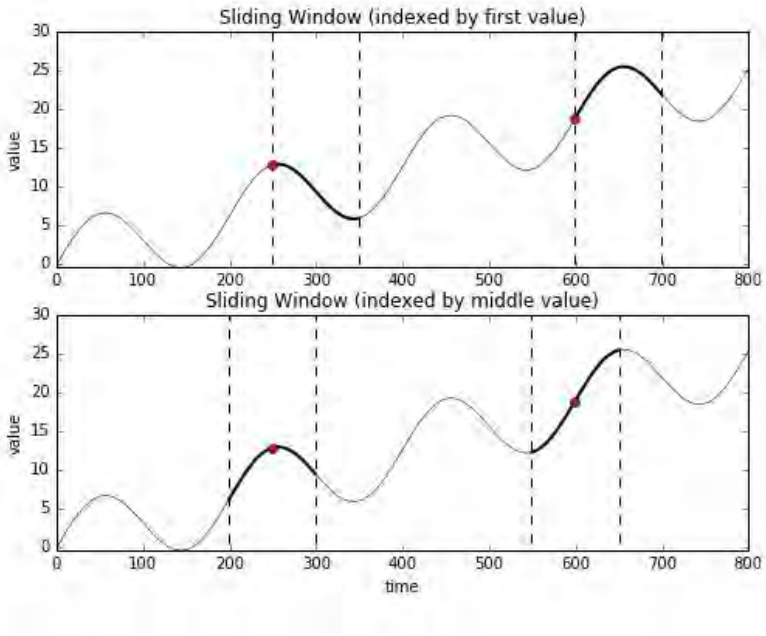
MEDIAN DIFFERENCE WINDOW SUBSERIES SCORE

In this chapter, a novel score for the contextual anomalies on the time series data is proposed. It is called Median Difference Window subseries Score or MDWS. In addition, the analysis of MDWS is presented along with the suggested thresholds and its algorithm.

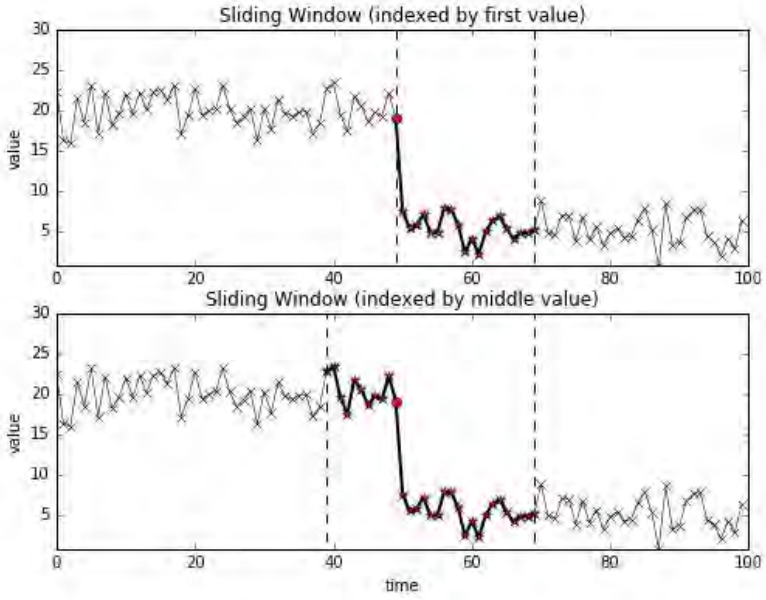
3.1 Definition of Median Difference Window Subseries Score

The definition of MDWS is presented in this section. The MDWS relies on the idea that each data point in the time series data should be associated with its surrounding context both preceding and succeeding. Then, two issues are considered which is, how to define the context of each data point and how to measure the distance between each data point with its context.

For the first issue, the sliding window (indexed by the middle value) is used, due to the center location of the surrouding data points. The distribution of the data points in two types of sliding window subseries is shown in Figure 3.1. Figure 3.1(a) presents the window subseries of two data points in the periodic time series on different locations: the local maximum and middle point in the period. Obviously, the sliding window indexed by the middle value shows the distribution of the context around each data point closer to the actual distribution than the sliding window indexed by the first value. In the case of the time series which contains more than one distribution, such as in Figure 3.1(b), the sliding window which is indexed by the first value is hard to capture the actual distribution of a joint point. Nevertheless, the sliding window which is indexed by the middle value still contains the data points that are generated from the same distribution



(a) The sliding window of the periodic time series data.



(b) The sliding window of the time series data which contains two distributions.

Figure 3.1**:** The distribution of the context around some data points in the sliding window subseries of two examples.

of the middle-window point more than half of the total.

For the distance between a point and a nonempty set, the standard distance is

frequently used. It is the minimum distance between the point and all elements in the set. Mathematatically, the distance between the point x and the nonempty set A is defined as:

$$d(x, A) = min\{d(x, a)|a \in A\}$$

where d is the distance function. In univariate dataset the distance function is defined by $d(x, y) = |x - y|$. Apparently, if $x \in A$ then $d(x, A) = 0$. Hence, this measure is not suitable for measuring the distance between the middle-window point with its window subseries. Then, the measuring distance between the middle-window point and the point is used to represent the nature of its window subseries. The central tendency which does not change suddenly with the influence of anomalies, i.e. the median is used for representing the window subseries. Consequently, the distance between the middle-window point y and its window W is defined as:

$$d(y, W) = |y - \widetilde{W}|$$

where $\widetilde{W}$ is the median of W. However, this thesis interests in the positive and negative value with respect to the data points, then the distance is redefined as follows:

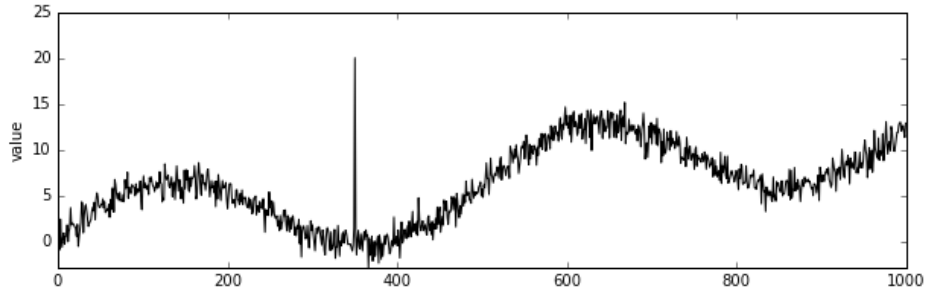
$$d(y, W) = y - \widetilde{W}$$

Next, the formal definition of Medain Difference Window subseries Score (MDWS) is defined by the following definition:

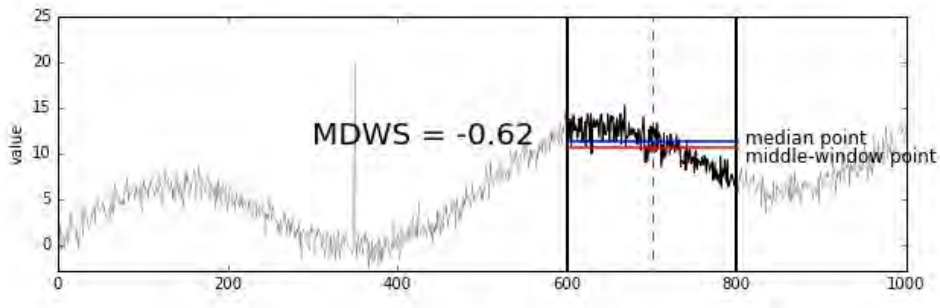
Definition 3.1**.** Given a time series $Y = \{y_0, y_1, ..., y_{n-1}\}$ of size n and a sliding window subseries (indexed by middle value) W_i^Y with length $2k+1$ of y_i for $i = k, k+1, ..., n-k-1$, the MDWS of y_i is the subtraction between y_i and the median of W_i^Y, i.e.

$$MDWS(y_i) = y_i - \widetilde{W_i^Y}.$$

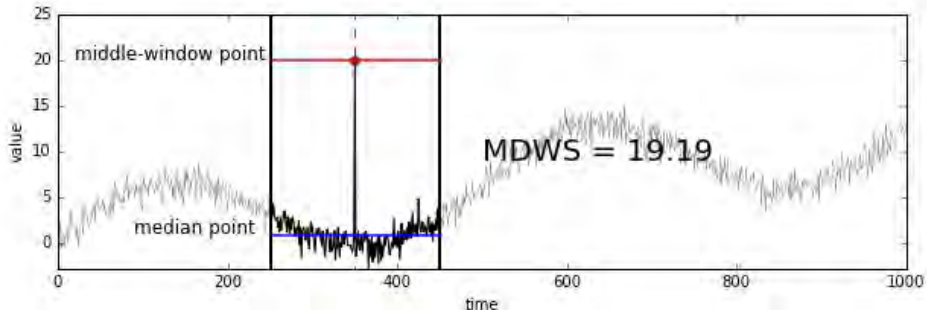
The calculation of MDWS is shown in Figure 3.2. The considered time series data is shown in Figure 3.2(a), then the computation of MDWS of a normal data point and anomaly are presented in Figure 3.2(b) and Figure 3.2(c) respectively.



(a) Dataset



(b) the MDWS of normal data point



(c) the MDWS of anomaly

Figure 3.2**:** The example of Median Difference Window subseries Score.

Figure 3.2 shows that the MDWS of the normal data point is near zero. On the other hand, the MDWS of the anomaly is very large, because it is very far from its context which represents by the median of its window subseries.

The next example shows the MDWS which set the parameter k to 50 in the time

series data of length 1000 which containes 2 anomalies is shown in Figure 3.3. The MDWS can distinguish the anomalies out of the normal data points. Only 900 values of MDWS are obtained due to the selection of the middle value, that is the MDWS of the begining and the end of the time series will be ignored.

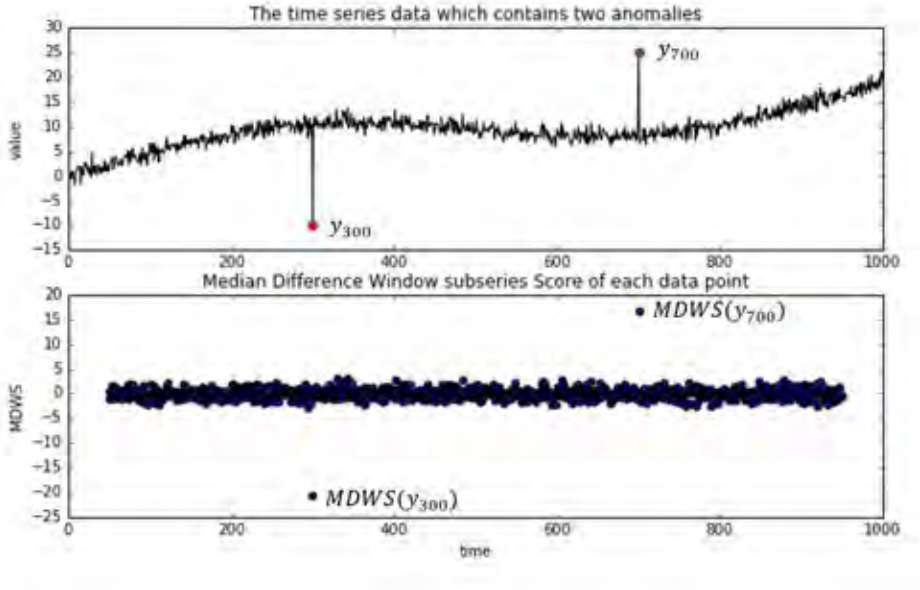


Figure 3.3**:** The MDWS of each data point in the time series data.

3.2 Analysis of MDWS

The MDWS is designed to perform effectively on the time series without trend and seasonal components. However, the real time series normally contains these two components. In this section, the effect of these two components with respect to MDWS are analyzed.

Effect of Trend Component

The effect of monotonic trend in each window subseries that affects MDWS is analyzed in this section. First, considering the time series without trend components with an anomaly (Figure 3.4(a)), the normal data point has small MDWS and the anomaly has high MDWS. Then, the time series is disturbed by linear trend (Figure 3.4(b)) and quadratic trend (Figure 3.4(c)). MDWS of the normal data point and anomaly are still

small close to the original value. That is because the median is robust against the effect of trend. Therefore, MDWS does not affect by the trend component.

Effect of Seasonal Component

The seasonality appears in the form of periodic function. When the window size is too large, the median of window subseries may be far away from the middle-window point, especially the turning points both the maximum and the minimum. In the case of that middle-window point is normal, the value of the MDWS is large, see Figure 3.5(a). Especially, if the anomaly appears at the turning point, the MDWS may be small, see Figure 3.5(b) which may not identify this data point as anomaly.

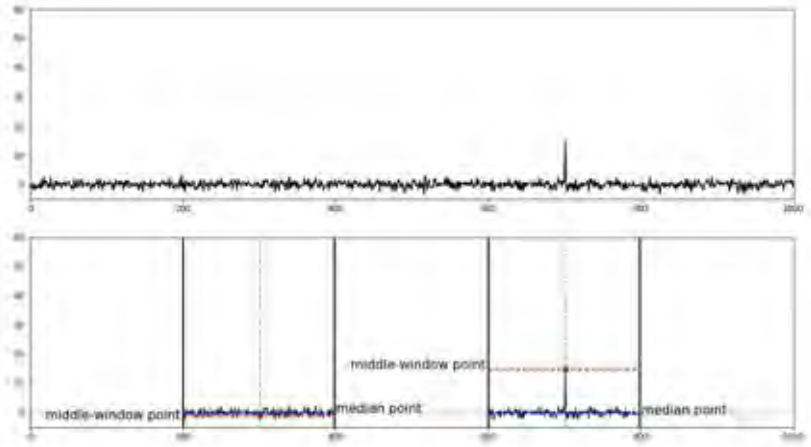
To avoid this situation, the window size should not be larger than a half of the period length such as the quarter of period length which is shown in Figure 3.5(c).

3.3 Suggested Thresholds

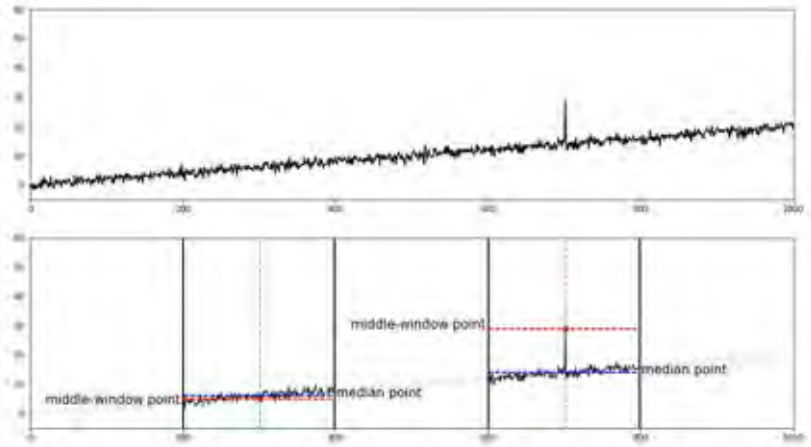
To identify the anomalies in non-temporal univariate dataset, two tails thresholds called interquartile range rule is used. It specifies the data point which is out of range between two statistics, the first quartile (Q_1) minus 1.5 times interquartile range (IQR) and the third quartile (Q_3) plus 1.5 times interquartile range (IQR), i.e. $[Q_1 - 1.5 \cdot IQR, Q_3 + 1.5 \cdot IQR]$ be anomaly (see Figure 3.6).

In some characteristics of time series data, most MDWSs are zero such as in Figure 3.7. If the interquartile range rule is used for identifying the anomalies in that dataset, then the upper and lower threshold are both zero causing all data points which their MDWS are not equal to zero to be specified as the anomalies.

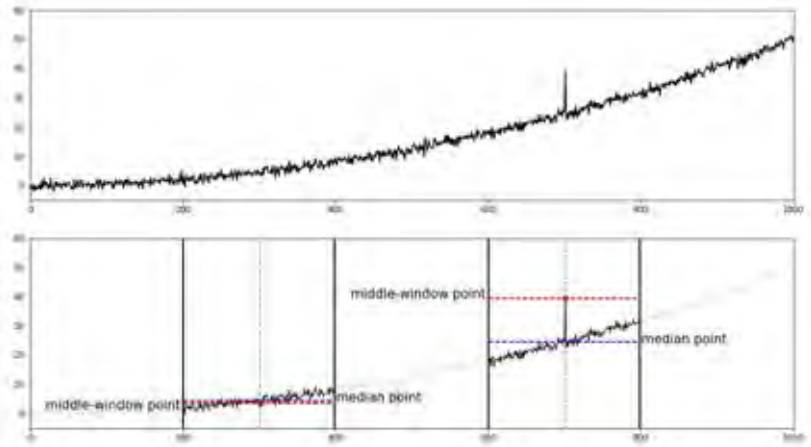
To avoid this problem, the leading thresholds are shown in Figure 3.8. It relies on the fact that the anomalies has very small number in a dataset. It considers the eighty percentage as the normal data points, i.e. the data points which have the MDWS in the range of D_1 the lower value to D_9 the upper value. For the data points having the MDWS



(a) No trend component

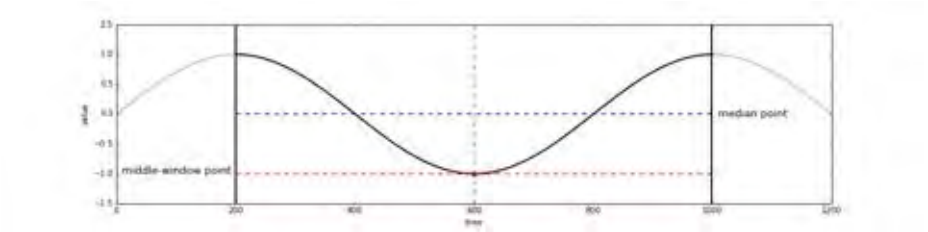


(b) Linear trend added

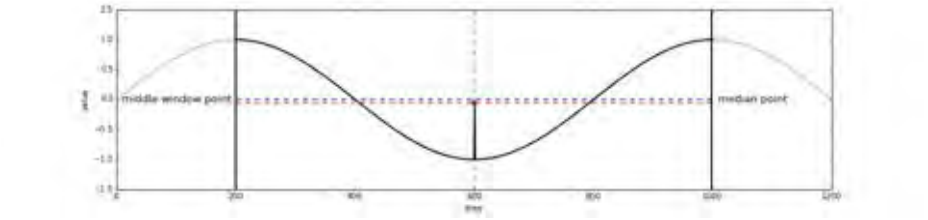


(c) Quadratic trend added.

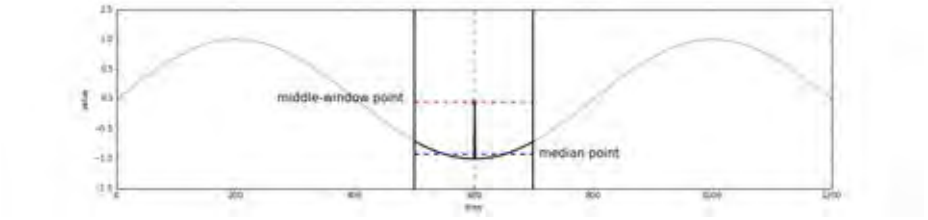
Figure 3.4**:** The effect of trend component.



(a) The case that the window size is large and the turning point is normal.



(b) The case that the window size is large and the turning point is anomaly.



(c) The case that the window size is small.

Figure 3.5**:** Seasonal component that affects MDWS at the turning point.

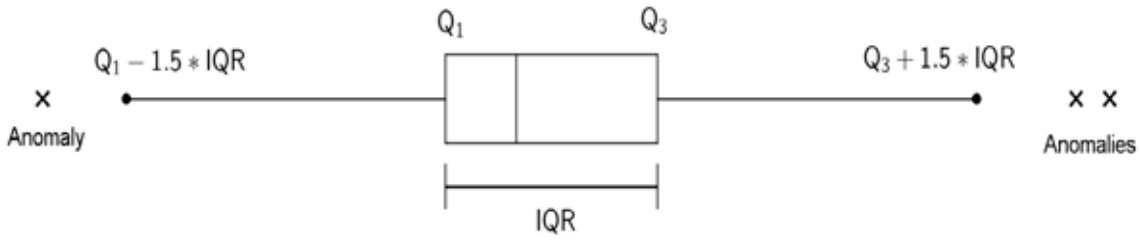


Figure 3.6**:** Interquartile Range Rule.

lower than D_1, the interquartile range rule is applied. This case considers only one tail in the left side. The interquartile range rule is used for the data point which is lower than $Q_1 - 3 * IQR$ to be assigned as an anomaly. Similarly, for the upper part, the data point

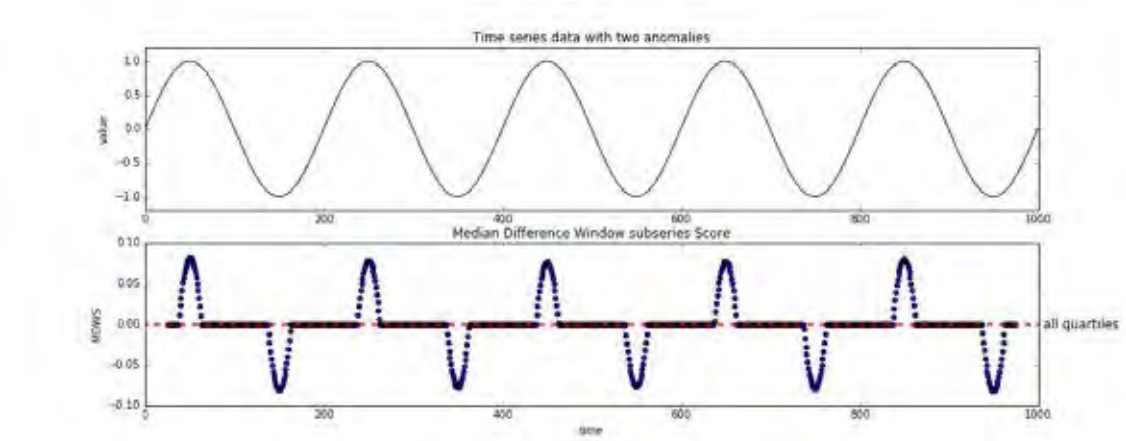


Figure 3.7**:** The MDWS of some time series data are zero more than a half.

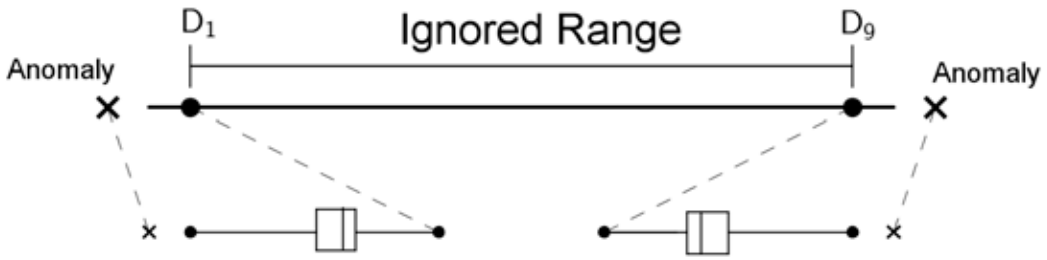


Figure 3.8**:** Suggested Thresholds.

which is greater than $Q_3 + 3 * IQR$ is specified to be an anomaly.

Three examples are presented in Figure 3.9, it show that our suggested thresholds are highly effective for separating the anomalies out of the normal data points.

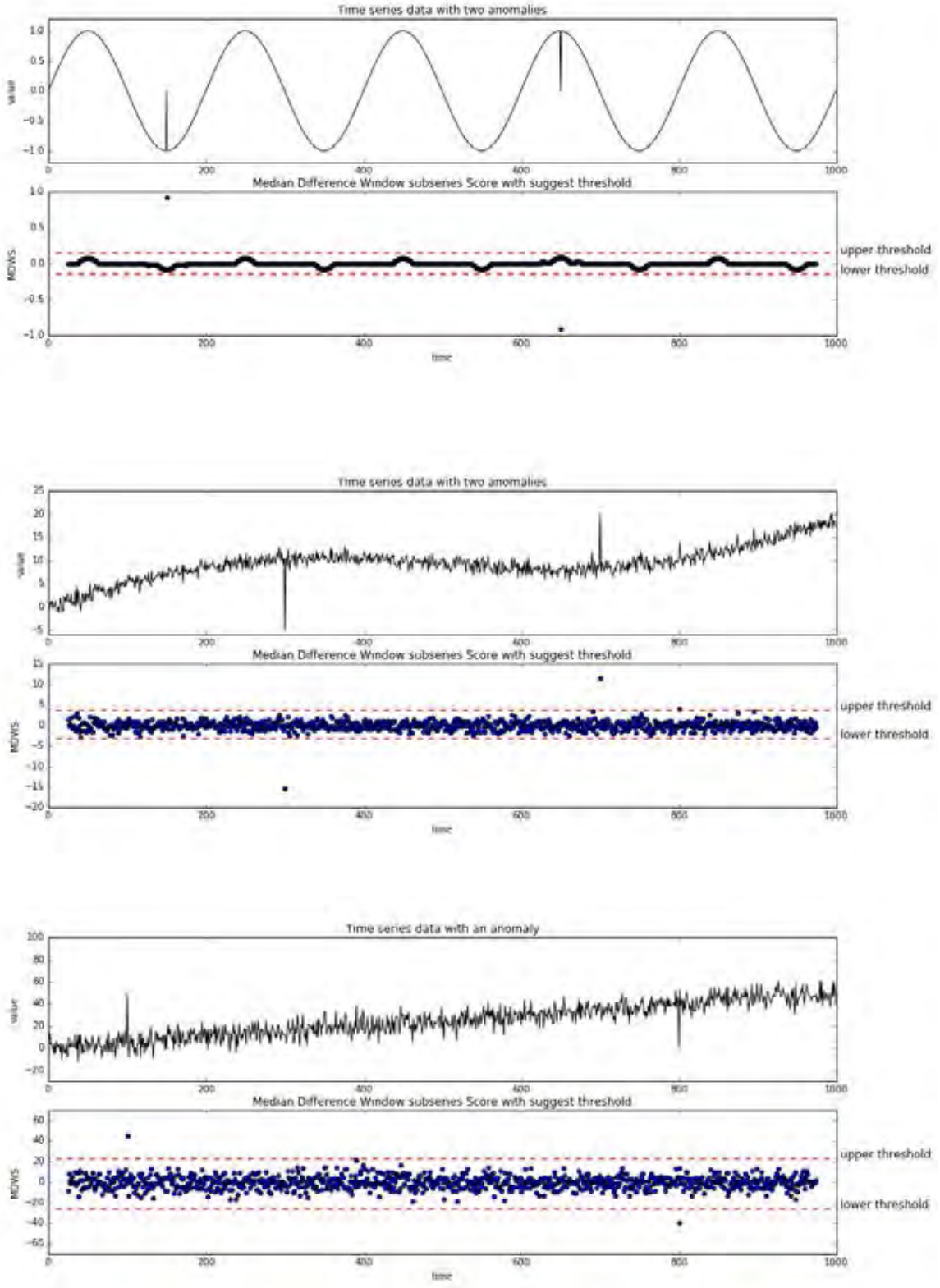


Figure 3.9**:** The suggested thresholds for MDWS.

3.4 Proposed Algorithm

In this section, two algorithms which are used in this thesis are proposed. The first algorithm is used for assigning the MDWS to each data point in a time series data. The second algorithm is presented for identifying the anomalies. Finally, the time complexity of these two algorithms are analyzed.

To minimize the time complexity for computing the MDWS, the MDWS algorithm (Algorithm 1) will perform an update from the entering data point. It finds the median point of the first window by sorting and updates the median point for the other windows by removing the expired data point and inserting the next data point. Then, computing the MDWS by subtracting the middle-window point with that median point.

Algorithm 1 MDWS Algorithm

Require: The time series $Y = \{y_0, y_1, ..., y_{n-1}\}$ of length n and the sliding window subseries (indexed by middle point) of size $2k + 1$.

1: let $MDWS = [\ ]$

2: sort the data points in the first window W_k, $W_{sorted} = sorted(W_k) = [s_0, ..., s_{2k}]$

3: pick the median point $\widetilde{W}_k = s_k$

4: calculate MDWS of y_k : $score(y_k) = y_k - \widetilde{W}_k$

5: append $score(y_k)$ into $MDWS$

6: keep W_{sorted}

for $i = k + 1, k + 2, ..., n - k - 1$ **do**

7: remove the data point y_{i-k-1} from W_{sorted}

8: insert the data point y_{i+k} in W_{sorted} and reordering index

9: pick the median point $\widetilde{W}_i = s_k$

10: calculate MDWS of y_i : $score(y_i) = y_i - \widetilde{W}_i$

11: append $score(y_i)$ into $MDWS$

keep W_{sorted}

end for

return $MDWS$

To identify a data point as an anomaly, the anomaly detection algorithm is presented in Algorithm 2. It uses the suggested thresholds in section 3.3 for separating the anomalies out of the normal data points.

Algorithm 2 Anomaly Detection Algorithm

Require: The set of MDWS, $MDWS = [score_1, score_2, ..., score_{n-2k}]$ with respect to the set of data points $[t_1, t_2, ..., t_{n-2k}]$

1: Let $Anomaly = \{\}$
2: Let D_1 and D_9 be the first and nineth decide of $MDWS$ respectively
3: $lower\_part = \{score_i \mid i = 0, ..., n-2k \ and \ score_i < D_1\}$
4: $upper\_part = \{score_i \mid i = 0, ..., n-2k \ and \ score_i > D_9\}$
5: Let Q_1^l and Q_3^l be the first and third quartile of $lower\_part$ respectively
6: Let Q_1^u and Q_3^u be the first and third quartile of $upper\_part$ respectively
7: $IQR^l = Q_3^l - Q_1^l$
8: $IQR^u = Q_3^u - Q_1^u$
9: $lower\_threshold = Q_1^l - 3 * IQR^l$
10: $upper\_threshold = Q_u^u + 3 * IQR^u$
for i = 1,...,$n-2k$ **do**
  if $score_i < lower\_threshold$ or $score_i > upper\_threshold$ **then**
    11: add t_i into $Anomaly$
  end if
end for
return $Anomaly$

Time Complexity Analysis

For the time complexity analysis, the MDWS algorithm is divided into two parts: 1) finding the median point and computing MDWS of the first window subseries and 2) updating the median point and returned the updated MDWS of the current window subseries. For the first part, the sorting process performs on $2k + 1$ elements which is the constant and compute the score, it takes $O(k \ log \ k)$ running time using the merge sort. In the second part, all windows use the insertion process and update the score, then this part uses $O(n \ log \ k)$ time complexity. The anomalous from the anomaly detection algorithm takes $O(n)$ running time. Then, the overall time complexity is $O(k \ log \ k) + O(n \ log \ k) + O(n) = O(n)$ running time.

CHAPTER IV

EXPERIMENTS AND RESULTS

The MDWS algorithm is implemented via Python programming language. All experiments in anomaly detection on time series data are presented in this section. The experiments are divided into two parts. First, the accuracy performance of the MDWS is shown compared with other techniques. Second, the efficiency of the MDWS algorithm is compared with FNWS algorithm.

4.1 Accuracy Performance

Firstly, the datasets which are used in this section are described. Next, the measures for testing the accuracy performance of each method are introduced. Then, the selection of each parameter is described. Finally, the core part of this section, i.e. the experiments with their results are shown.

Dataset

In order to test the performance of MDWS and other methods, the benchmark datasets from Yahoo! [32] and Numenta [20] are used. The benchmark from Yahoo! consists of four collections which are synthetic and real world time series data. For the benchmark from Numenta, it has five collections of the real world dataset. Concisely, Table 4.1 shows the summary of all datasets which are used in this thesis.

Measurements

In this section, three performance measures: Precision, Recall and F1-Measure are used for comparing the performance. Those measures can be derived from the values in

Source	Collection	Category	# Dataset	Data Length
Yahoo!	A1Benchmark	real world	67	741-1461
	A2Benchmark	synthetic	100	1421
	A3Benchmark	synthetic	100	1680
	A4Benchmark	synthetic	100	1680
Numenta	realAdExchange	real world	6	1538-1643
	realAWSCloudwatch	real world	17	1243-4730
	realKnownCause	real world	7	1882-22695
	realTraffic	real world	7	1127-2500
	realTweets	real world	10	15893-15902

Table 4.1**:** The collections of time series data which are used for testing the performance of MDWS and other methods.

the confusion matrix (see Table 4.2).

		Predicted	
		Anomaly	*Normal*
Actual	*Anomaly*	True Positive (TP)	False Negative (FN)
	Normal	False Positive (FP)	True Negative (TN)

Table 4.2**:** Confusion Matrix

The entries in the confusion matrix are TP, FP, FN, and TP:

- True Positive (TP) is the number of the data points that are predicted as anomalies, and they are actual anomalies.
- False Negative (FN) is the number of the data points that are predicted as normal data points, but they are anomalies.
- False Positive (FP) is the number of the data points that are predicted as anomalies, but they are normal data points.
- True Negative (TN) is the number of the data points that are predicted as normal data points, and they are actual normal points.

Three performance measures are presented as follows:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1-Measure} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

The precision is the percentage of correct predicted anomalies from the model. For the recall, it shows the proportion of the number of anomalies that are detected from the model. For example, if almost all predicted anomalies are correct, but there are many other anomalies are not detected, then the precision is high and the recall is low. On the other hand, if almost all anomalies are detected, but there are many incorrect predicted anomalies, then the precision is low and the recall is high. Finally, the F1-Measure is the harmonic mean of precision and recall. Then, the bigger F1-Measure value has the greater overall of precision and recall than the smaller F1-Measure.

Parameter Setting

The MDWS, FNWS and S-H-ESD are executed using the same window length. In the experiments, The window length is chosen to vary from from 1% to 20% of data length in each time series. Moreover, the FNWS algorithm requires an additional parameter k which is the k-nearest neighbors that set to be the same as the window length. For the S-H-ESD, a significant level is set to 0.05 and the number of maximum anomalies k be 2%.

4.1.1 Synthetic Dataset

The synthetic datasets used in this section are generated by Yahoo!. There are three collections of synthetic time series data, i.e. A2Benchmark, A3Benchmark and A4Benchmark. The description and some examples of each collection along with their experiments and results are presented in the next section.

Yahoo!/A2Benchmark

The collection A2Benchmark from the Yahoo! benchmark consists of 100 various synthetic time series data of length 1421 with the trend, seasonality, noises and anomaly tag labels. They are characterized by a periodic function like the sin function, see the example in Figure 4.1.

The experimental results are shown as the Figure 4.2. On the large window size, the MDWS (solid line) has very high precision close to 1 like S-H-ESD (dotdash line), because they identify only the exact anomalies. On the other hand, the FNWS (dashed line) has a lot of incorrect predicted anomalies. For the recall and F1-Measure, the performance of MDWS is better than the performance of FNWS which are the same with the large window length. The recall of S-HESD is worse than MDWS and FNWS in all window lengths similar to F1-Measure.

Due to the clarity of seasonality in each dataset of the collection A2Benchmark, the selection of window length is adjusted on the period length. Then, the window size is varied from 10% to 200% of the period length in each time series data.

The experimental results of the cases above are shown as the Figure 4.3. It shows that, when the suggested parameter for MDWS which is presented in the section 3.2 (around the quarter to the half of period length) is used, the MDWS shows the satisfactory precision, recall and F1-measure. More importantly, it has the best recall and F1-Measure among other parameters and other methods.

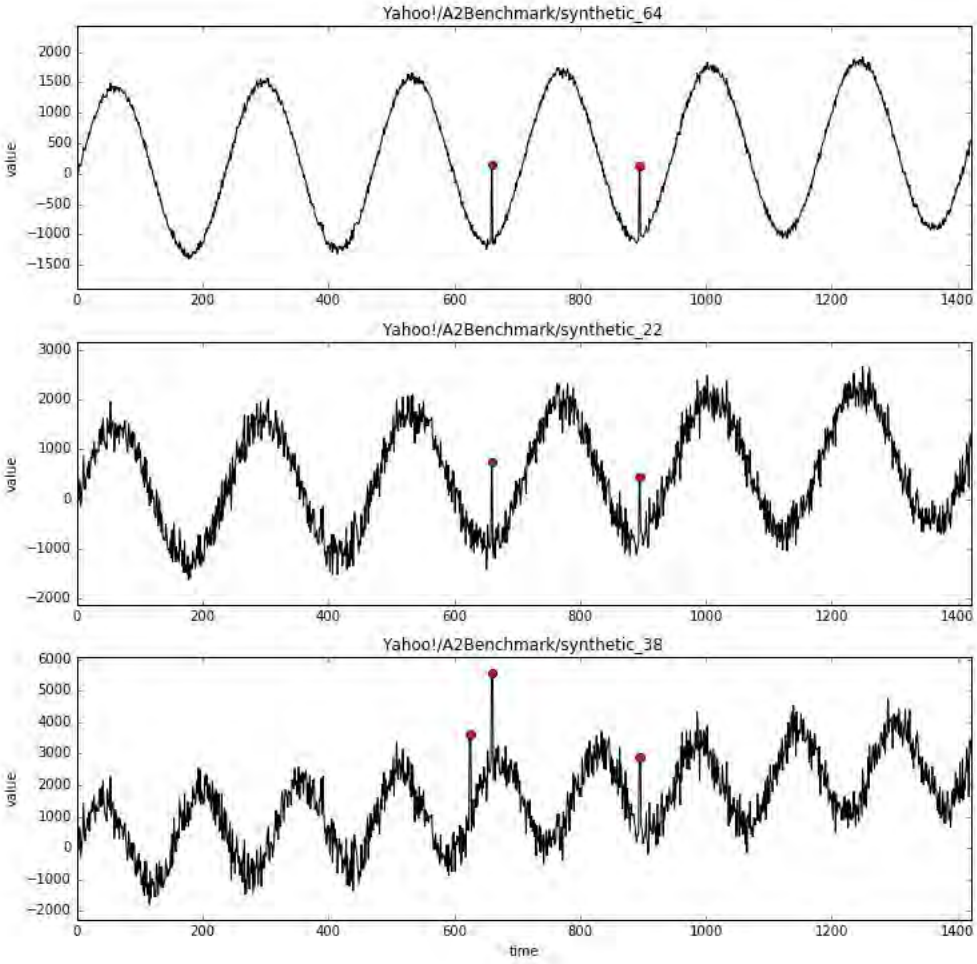


Figure 4.1**:** Three examples of time series data in the collection A2Benchmark from the Yahoo! benchmark.

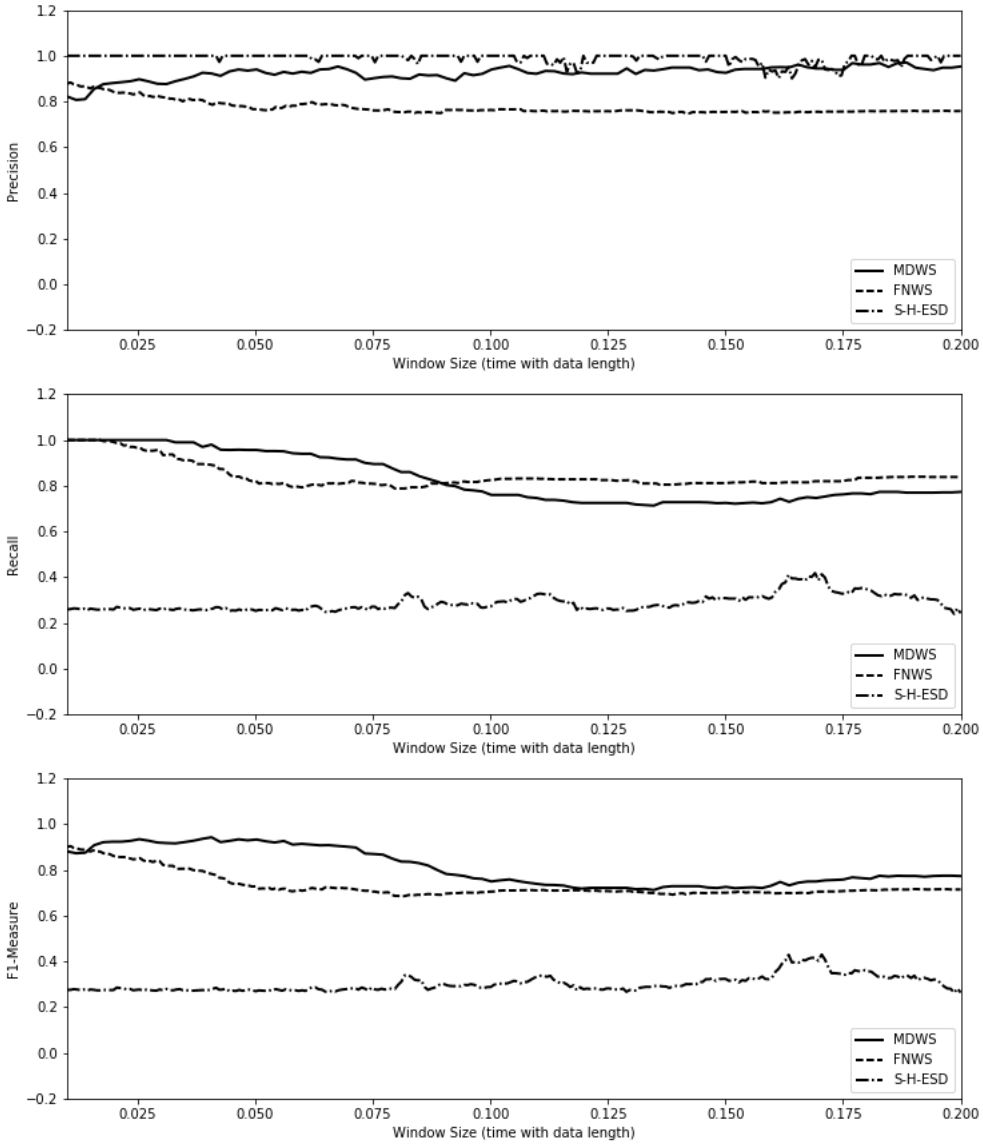


Figure 4.2**:** The performance of MDWS, FNWS and S-H-ESD to detect anomalies on the synthetic time series data in the collection A2Benchmark from the Yahoo! benchmark.

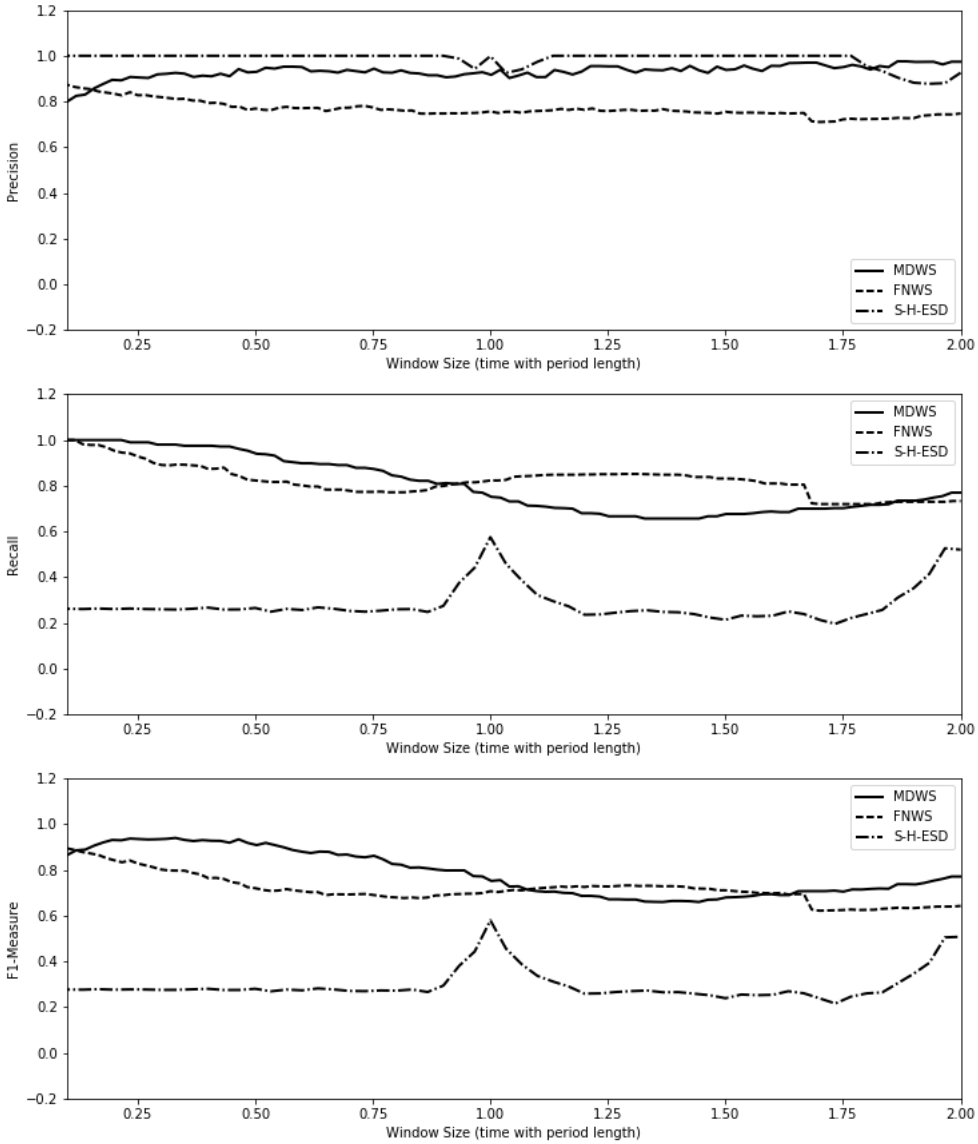


Figure 4.3**:** The performance of MDWS, FNWS and S-H-ESD to detect anomalies on the synthetic time series data in the collection A2Benchmark from the Yahoo! benchmark, when the window length varies on the period length.

Yahoo!/A3Benchmark

The collection A3Benchmark from the Yahoo! benchmark consists of 100 various synthetic time series data of length 1680 with the anomaly tag labels. There are abruptly change up and down around the trend and seasonality along with some noises, see the example in Figure 4.4.

The experimental results are shown as the Figure 4.5. The MDWS (solid line) has very high precision close to 1 in almost window length like S-H-ESD (dotdash line), because they identify only the exact anomalies. On the other hand, the FNWS (dashed line) has a lot of incorrect predicted anomalies. Moreover, the recall of MDWS is smaller than FNWS when the window size is large. For the small window length, the recall of MDWS is similar to FNWS. The recall of S-HESD is very low close to zero which is worse than MDWS and FNWS in all window length. However, the F1-Measure of MDWS still is the best.

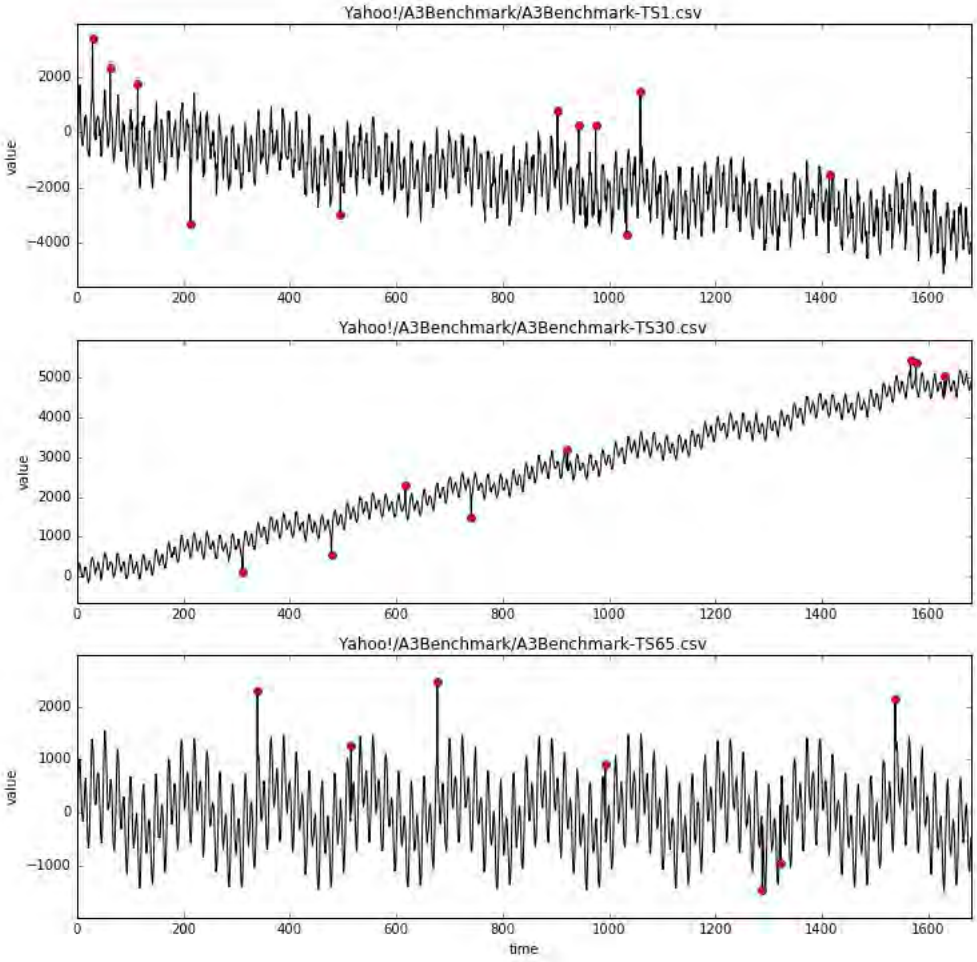


Figure 4.4**:** Three examples of time series data in the collection A3Benchmark from the Yahoo! benchmark.

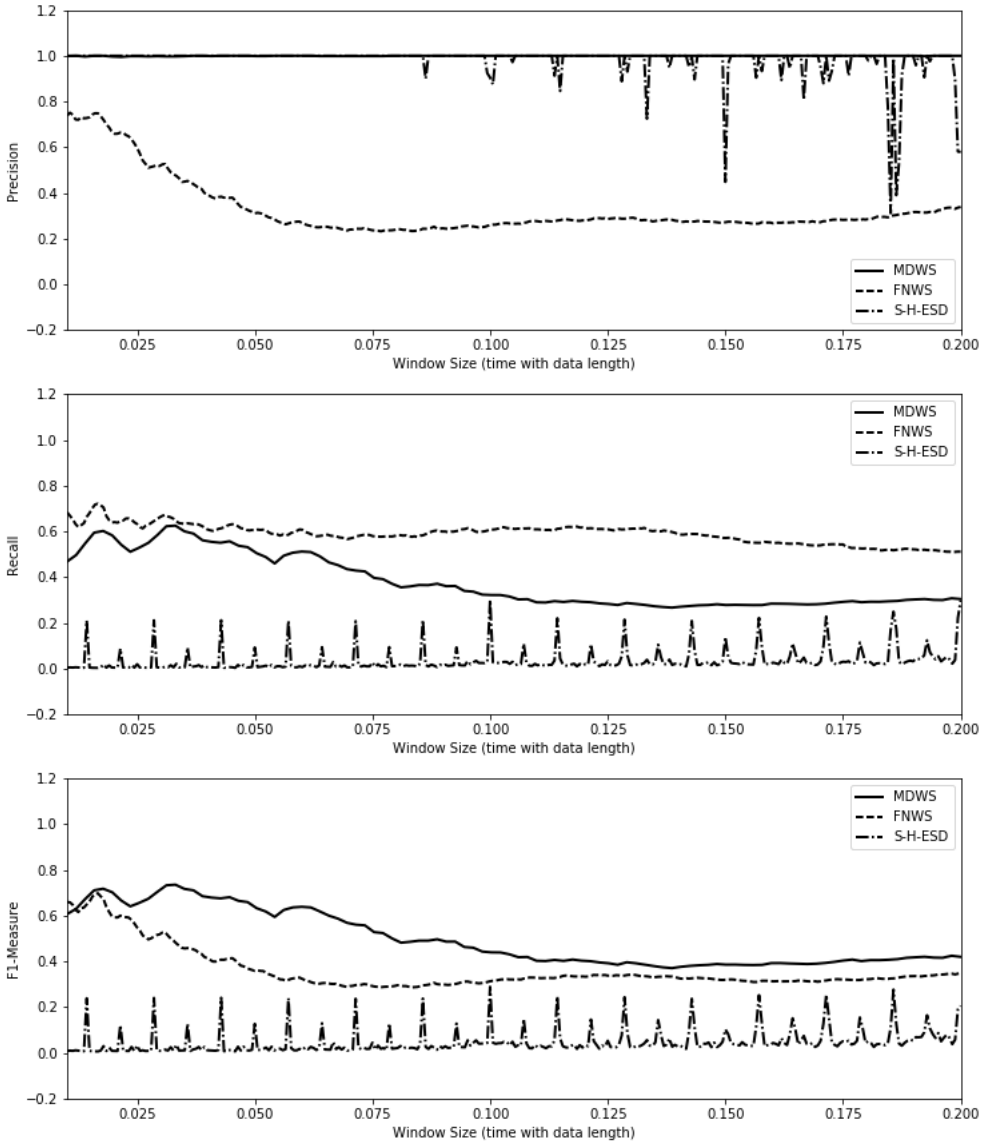


Figure 4.5**:** The performance of MDWS, FNWS and S-H-ESD to detect anomalies on the synthetic time series data in the collection A3Benchmark from the Yahoo! benchmark.

Yahoo!/A4Benchmark

The collection A4Benchmark from the Yahoo! benchmark consists of 100 various synthetic time series data of length 1680 with the anomaly tag labels. There are abruptly change up and down around the trend and seasonality along with some noises, like the collection A3Benchmark. Specially, each time series data in the collection A4Benchmark contains more than one structure which are generated from different distribution, see the example in Figure 4.6.

The experimental results are shown as the Figure 4.7. Since the MDWS is not designed for multi-distribution time series data, then the performance of MDWS (solid line) is low, like other methods. However, the precision of MDWS is still high, because it still identify only the explicit anomalies. On the other hand, many anomalies are not detected, then the recall is low causing F1-Measure is low too. Then, this situation suggest to consider each distribution separately. When this strategy is used, the performance of MDWS (asterisk line) is similar to the result on the collection A3Benchmark which is highest comparable with FNWS (dashed line) and S-H-ESD (dotdash line).

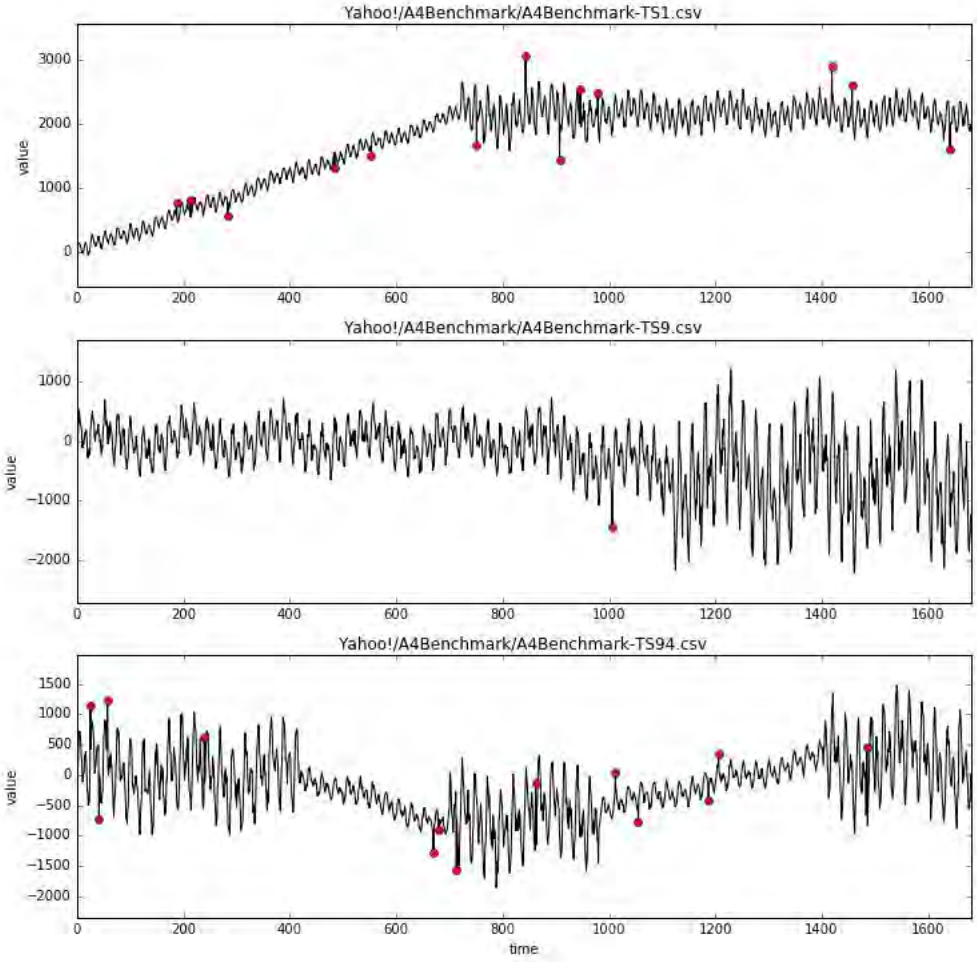


Figure 4.6**:** Three examples of time series data in the collection A4Benchmark from the Yahoo! benchmark.

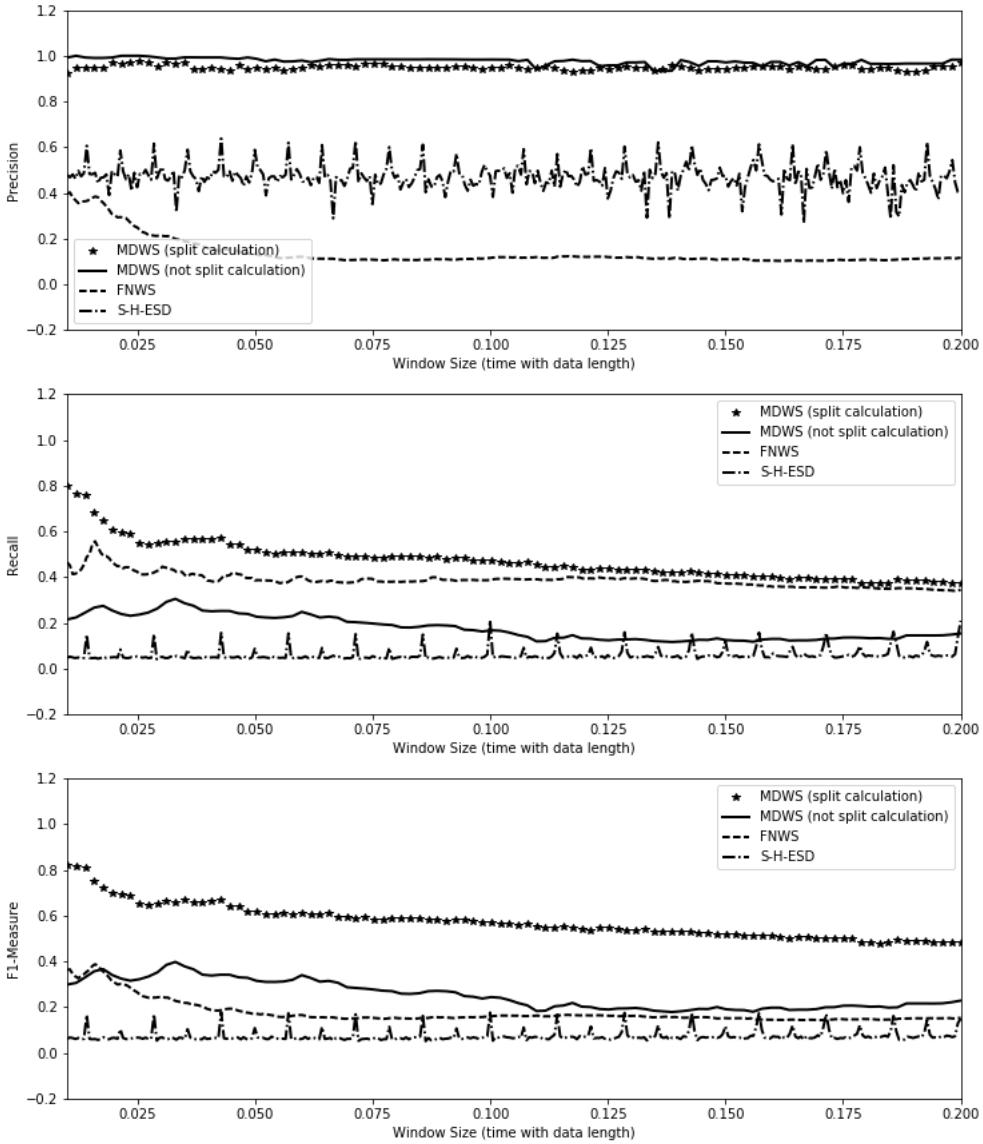


Figure 4.7**:** The performance of MDWS, FNWS and S-H-ESD to detect anomalies on the synthetic time series data in the collection A4Benchmark from the Yahoo! benchmark.

4.1.2 **Real World Dataset**

The real world datasets which are used in this thesis are provided from two sources, i.e. Yahoo! and Numenta. There is a collection A1Benchmark from the Yahoo! and five collections from the Numenta, i.e. realAdExchange, realAWSCloudwatch, realKnown-Cause, realTraffic and realTweets. The anomalies in each real world time series data are marked by humans and therefore may not be consistent.

Due to the variety and uncontrollability of each time series data, the characteristics of both the normal data points and the anomalies are not often based on assumption of MDWS. For example, some anomalies appear at the beginning or the end of time series data, and also appear as collective anomalies in some datasets. The performance of MDWS (solid line) is low and unpredictable. However, overall performances of MDWS is still better than the performance of FNWS (dashed line) and S-H-ESD (dotdash line).

The description and some examples of each collection along with their experiments and results are presented in the next order.

Yahoo!/A1Benchmark

The collection A1Benchmark from the Yahoo! benchmark consists of 67 real world time series data of length 741 to 1461 with anomaly tag labels. It is based on the real production traffic to some of the Yahoo! properties, see the examples in Figure 4.8.

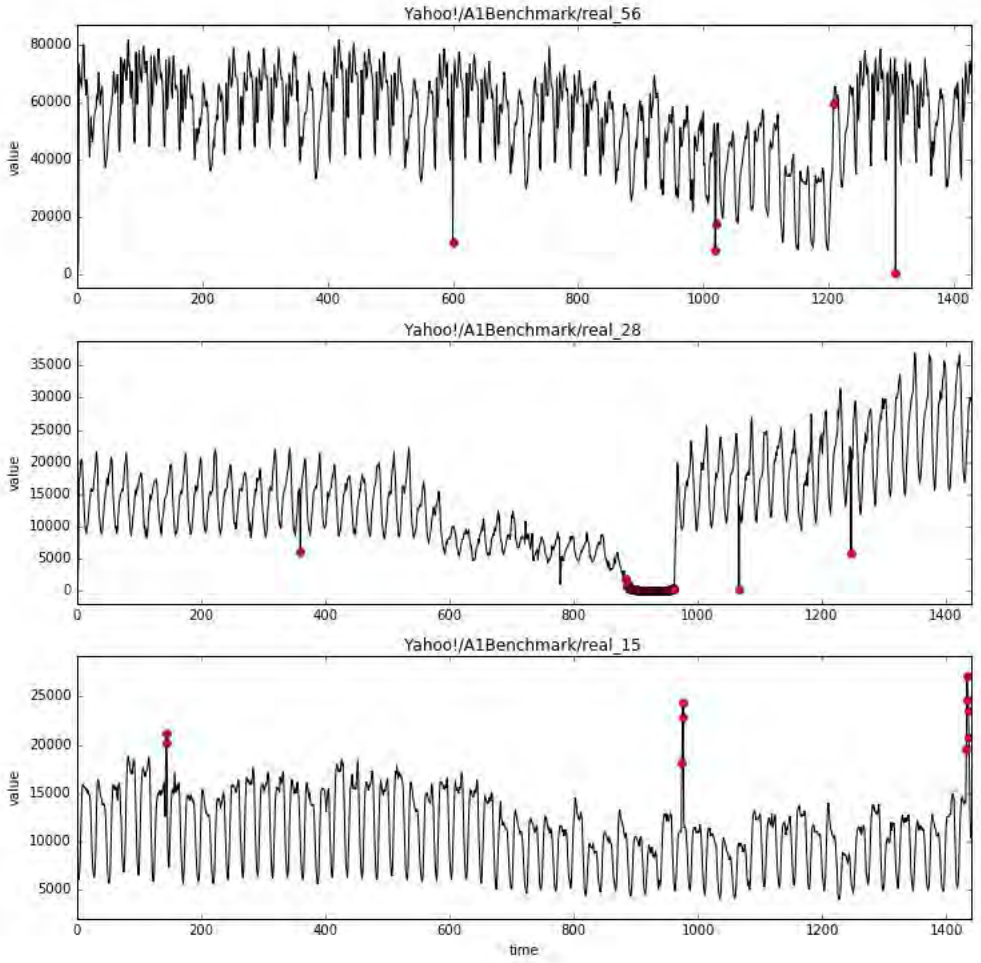


Figure 4.8**:** Three examples of time series data in the collection A1Benchmark from the Yahoo! benchmark.

The experimental results for the performance of each time series data in the collection A1Benchmark are shown as the Figure 4.9 on the next page.

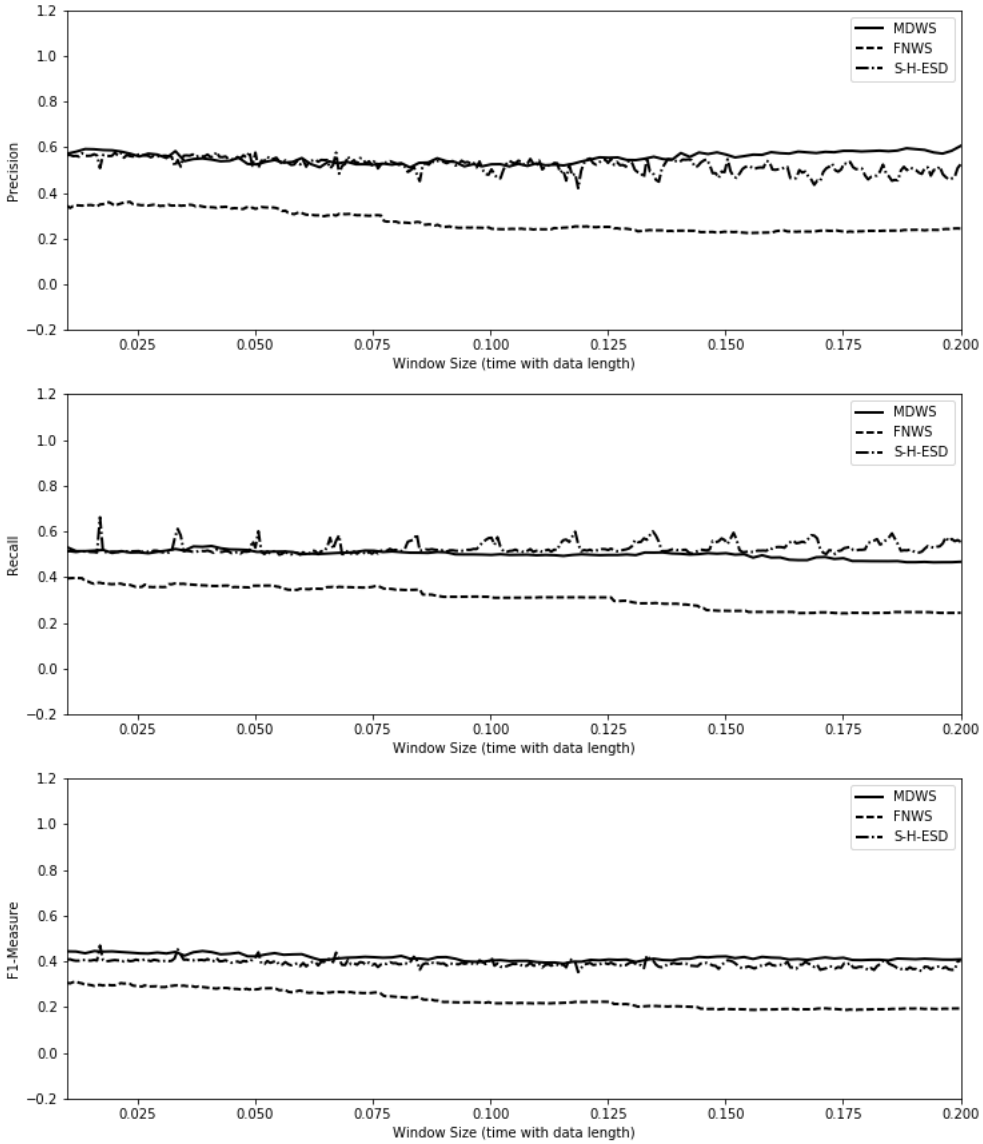


Figure 4.9**:** The performance of MDWS, FNWS and S-H-ESD to detect anomalies on the real world time series data in the collection A1Benchmark from the Yahoo! benchmark.

Numenta/realAdExchange

The collection realAdExchange from the Numenta benchmark consists of 6 real world time series data of length 1538 to 1643 with the anomaly tag labels. It consists of the online advertisement clicking rates, where the metrics are cost-per-click (CPC) and cost per thousand impressions (CPM), see the examples in Figure 4.10.

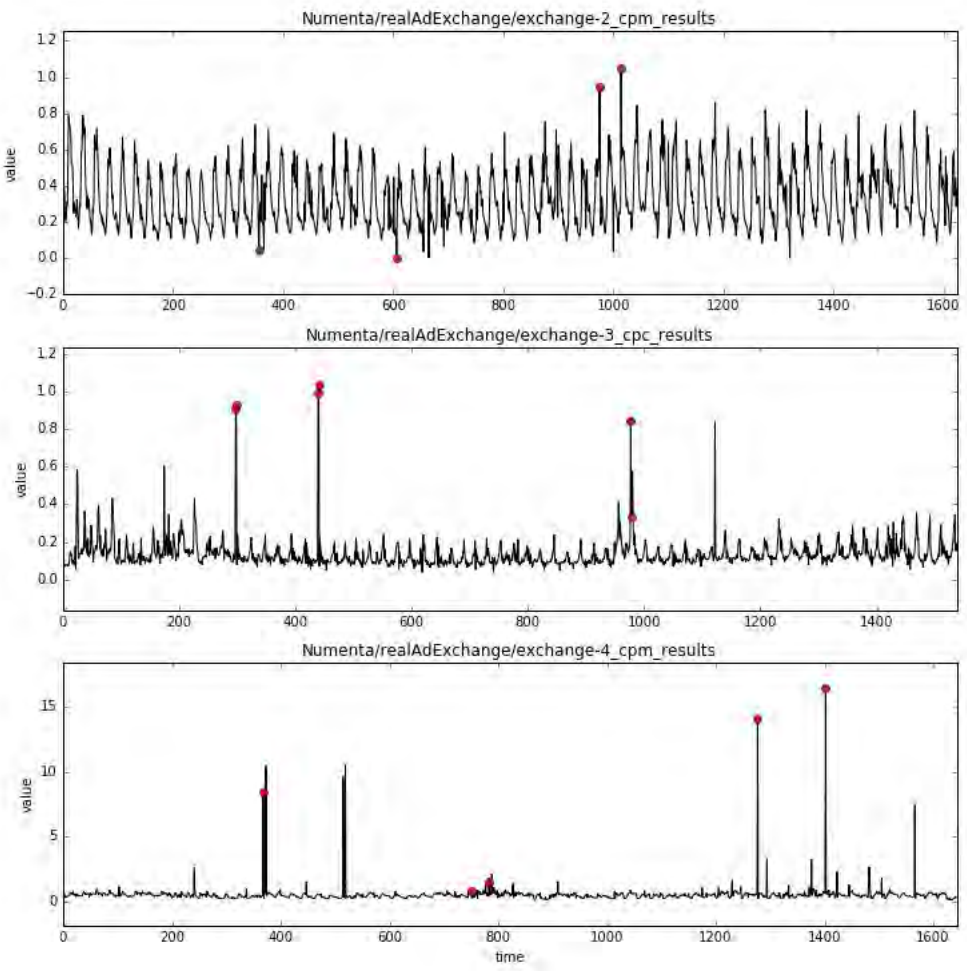


Figure 4.10**:** Three examples of time series data in the collection realAdExchange from the Numenta benchmark.

The experimental results for the performance of each time series data in the collection realAdExchange are shown as the Figure 4.11 on the next page.

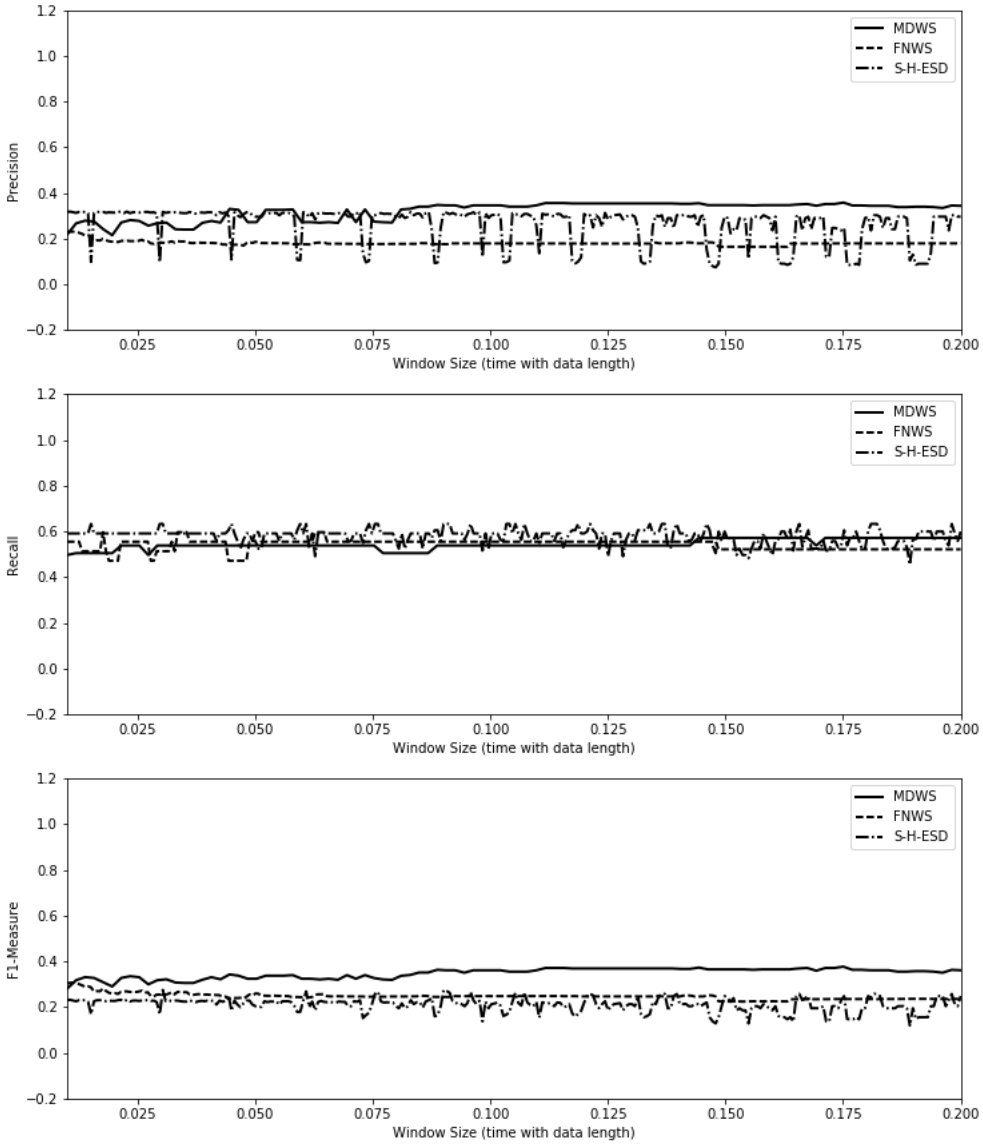


Figure 4.11**:** The performance of MDWS, FNWS and S-H-ESD to detect anomalies on the real world time series data in the collection realAdExchange from the Numenta benchmark.

Numenta/realAWSCloudwatch

The collection realAWSCloudwatch from the Numenta benchmark consists of 17 real world time series data of length 1243 to 4730 with the anomaly tag labels. It consists of the AWS server metrics as collected by the AmazonCloudwatch service, see the examples in Figure 4.12.

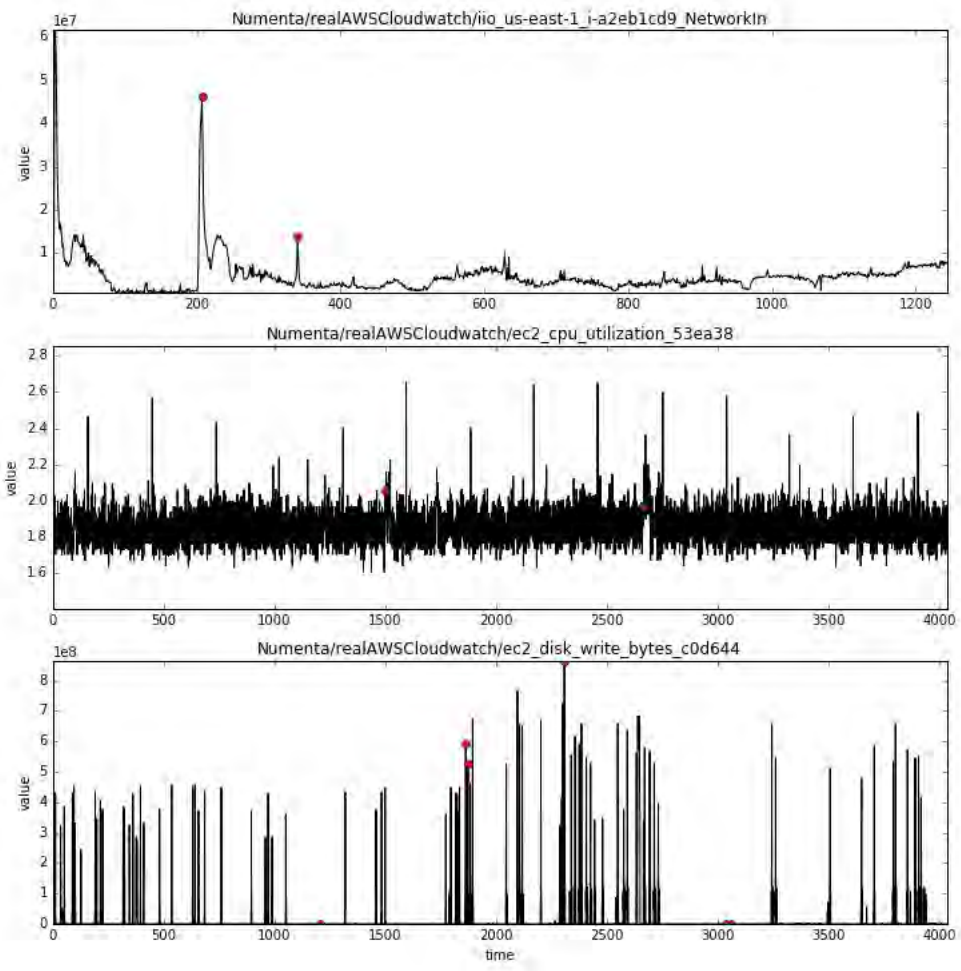


Figure 4.12**:** Three examples of time series data in the collection realAWSCloudwatch from the Numenta benchmark.

The experimental results for the performance of each time series data in the collection realAWSCloudwatch are shown as the Figure 4.13 on the next page.

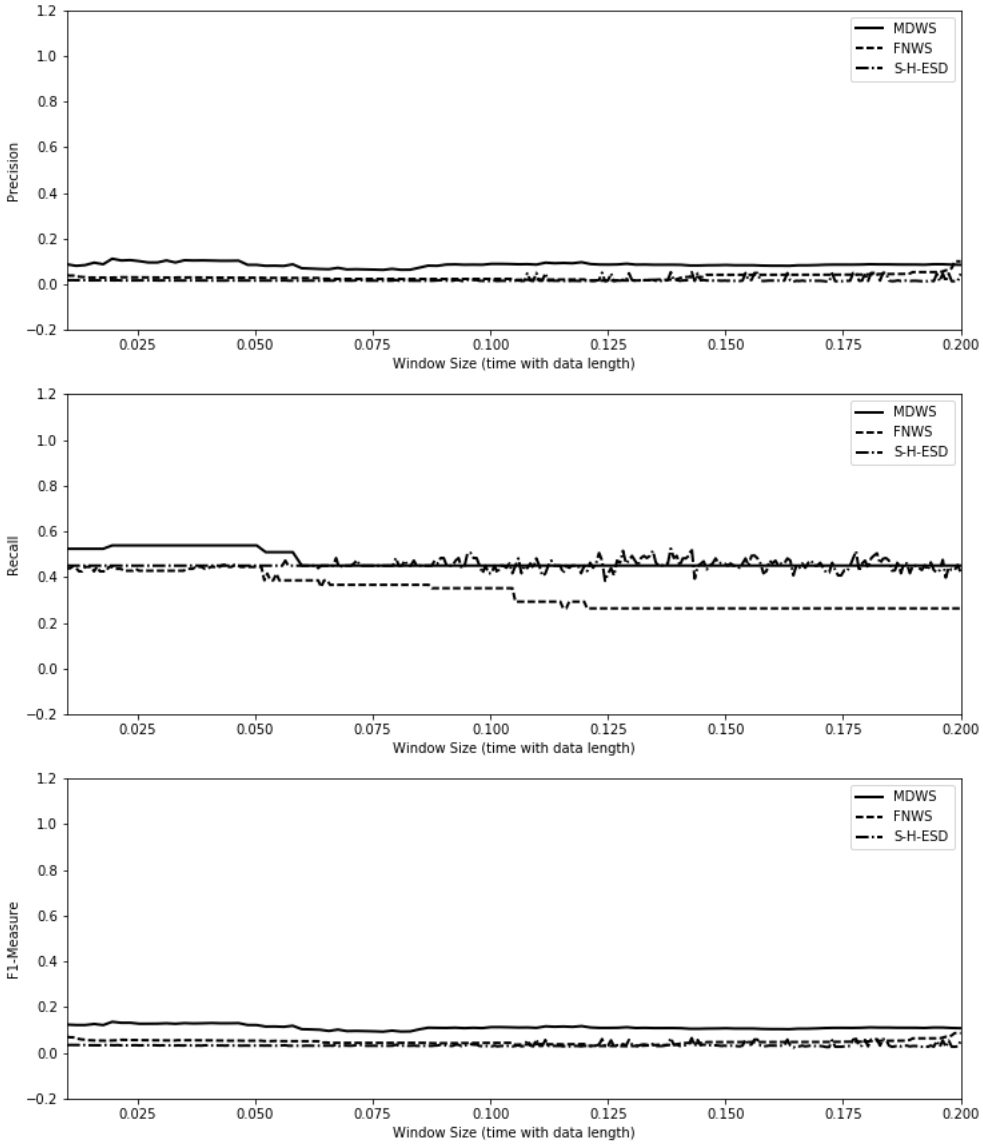


Figure 4.13**:** The performance of MDWS, FNWS and S-H-ESD to detect anomalies on the real world time series data in the collection realAWSCloudwatch from the Numenta benchmark.

Numenta/realKnownCause

The collection realKnownCause from the Numenta benchmark consists of 7 real world time series data of length 1882 to 22695 with the anomaly tag labels. Especially, each dataset in this collection is known the anomaly causes; no hand labeling, see the examples in Figure 4.14.

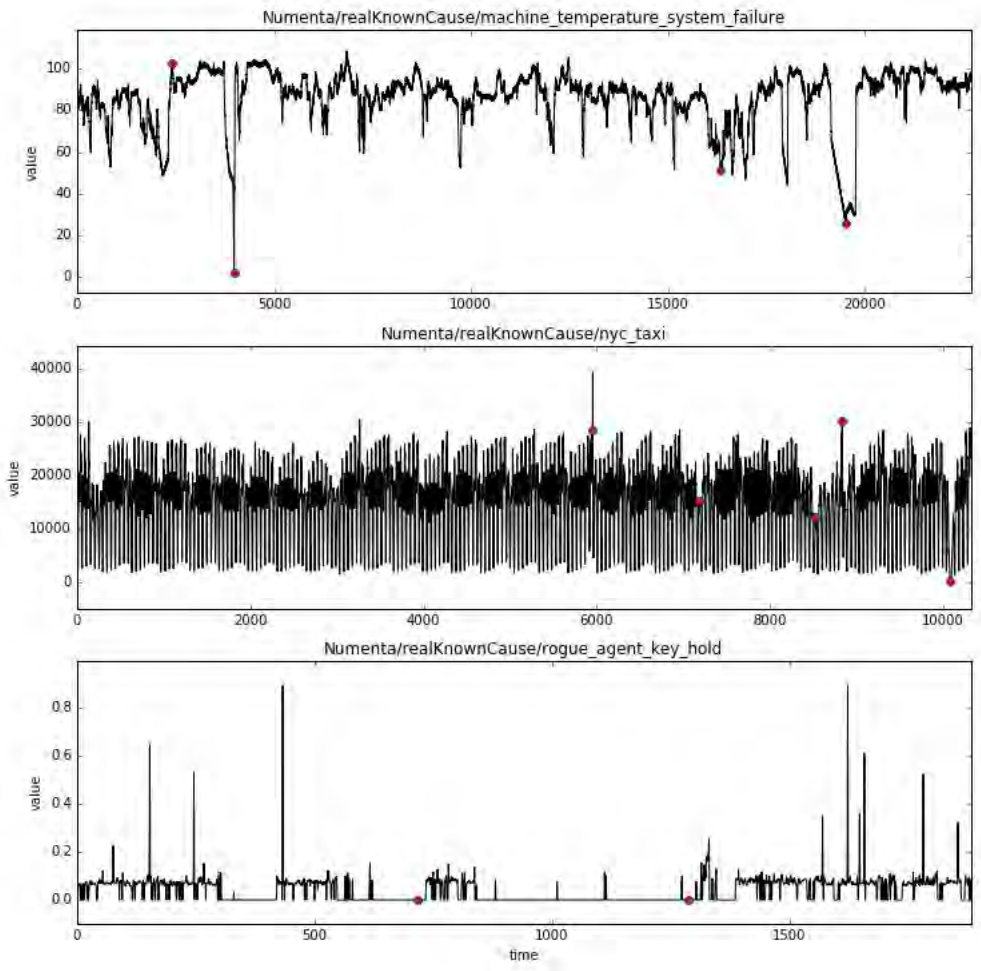


Figure 4.14**:** Three examples of time series data in the collection realKnownCause from the Numenta benchmark.

The experimental results for the performance of each time series data in the collection realKnownCause are shown as the Figure 4.15 on the next page.

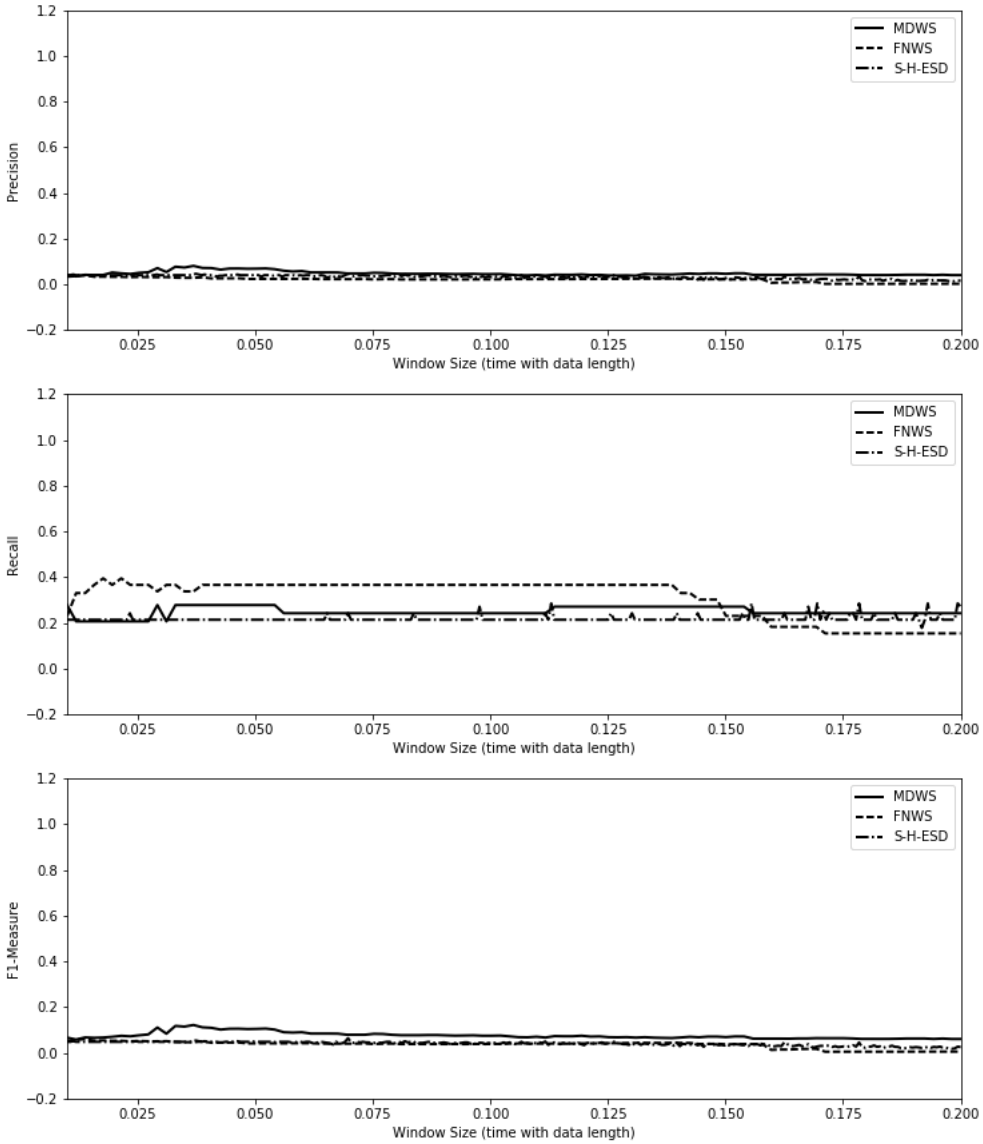


Figure 4.15**:** The performance of MDWS, FNWS and S-H-ESD to detect anomalies on the real world time series data in the collection realKnownCause from the Numenta benchmark.

Numenta/realTraffic

The collection realTraffic from the Numenta benchmark consists of 7 real world time series data of length 1127 to 2500 with the anomaly tag labels. It is provided from the Twin Cities Metro area in Minnesota, collected by the Minnesota Department of Transportation, see the examples in Figure 4.16.

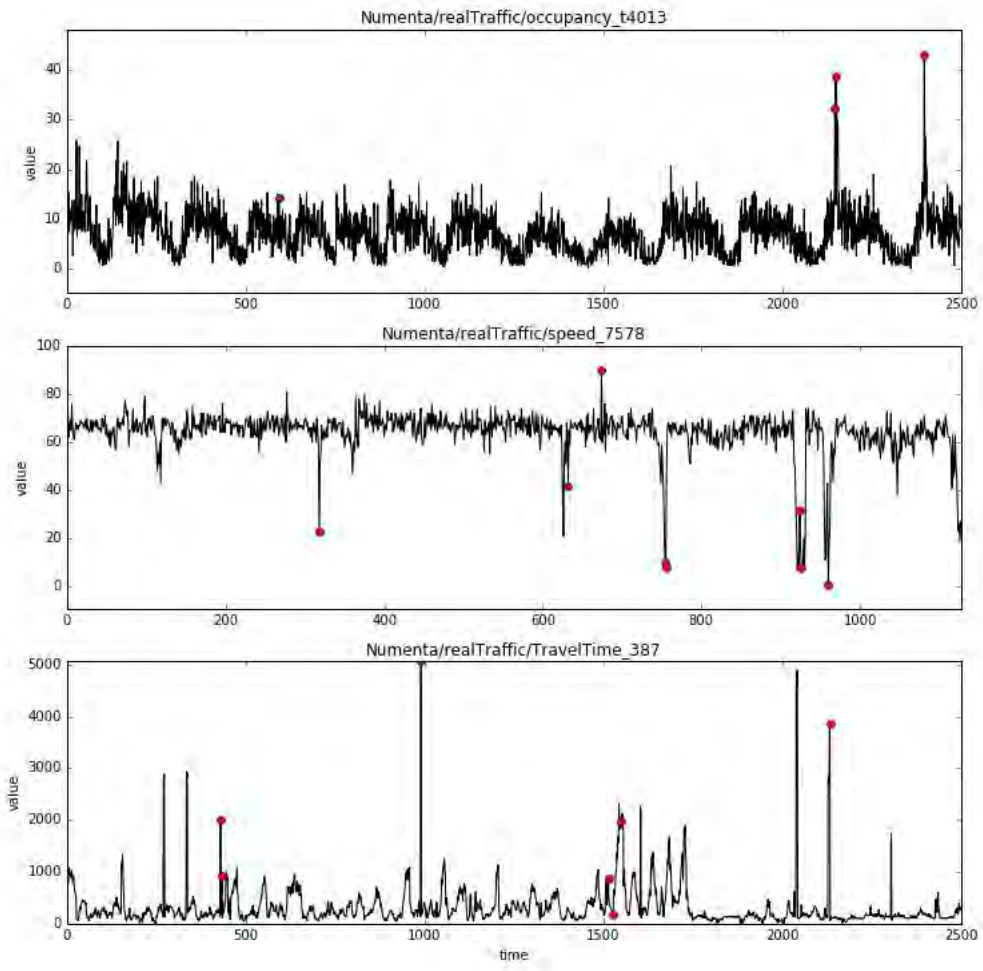


Figure 4.16**:** Three examples of time series data in the collection realTraffic from the Numenta benchmark.

The experimental results for the performance of each time series data in the collection realTraffic are shown as the Figure 4.17 on the next page.

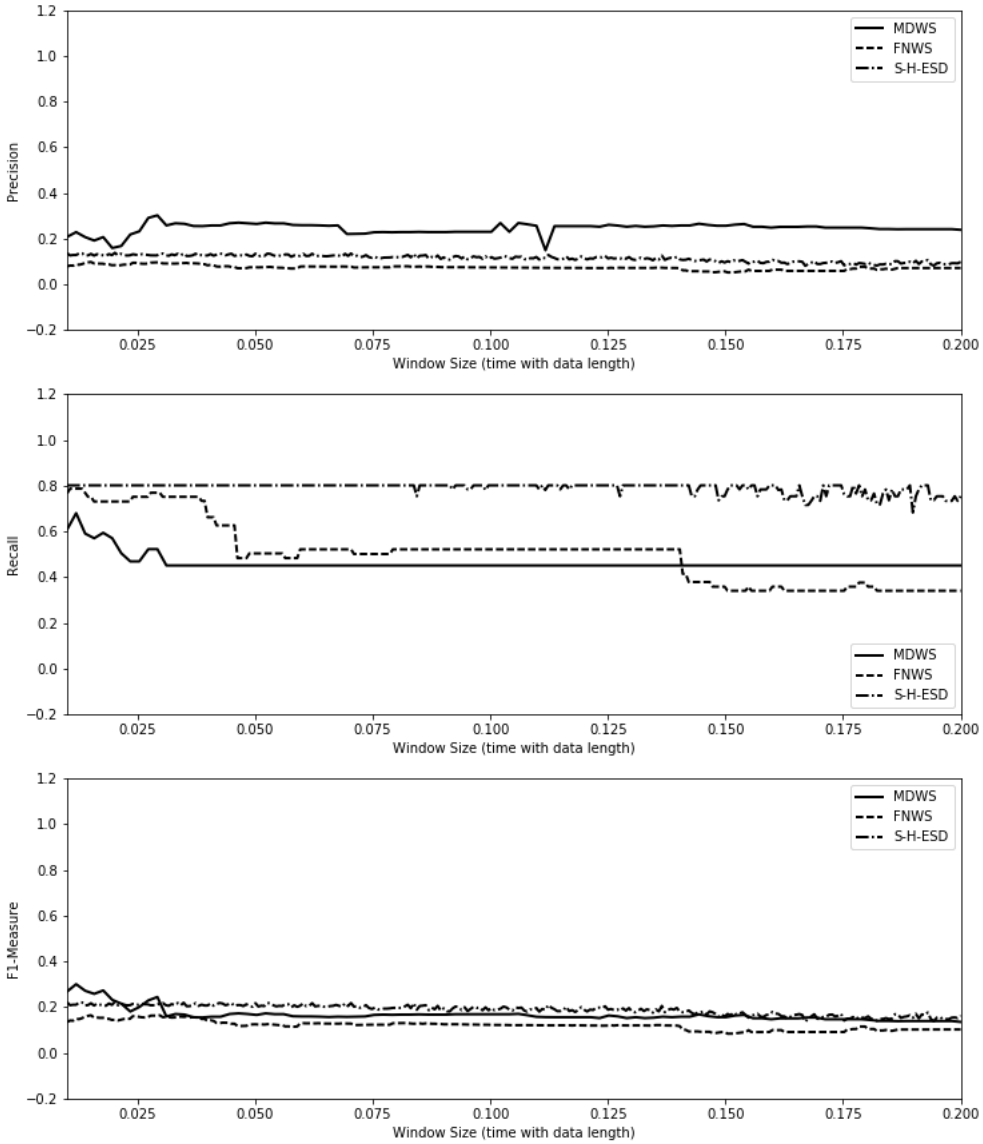


Figure 4.17**:** The performance of MDWS, FNWS and S-H-ESD to detect anomalies on the real world time series data in the collection realTraffic from the Numenta benchmark.

Numenta/realTweets

The collection realTweets from the Numenta benchmark consists of 10 synthetic time series data of length 15893 to 15902 with the anomaly tag labels. It is a collection of Twitter mentions of large publicly-traded companies such as Google and IBM, sew the examples in Figure 4.18.

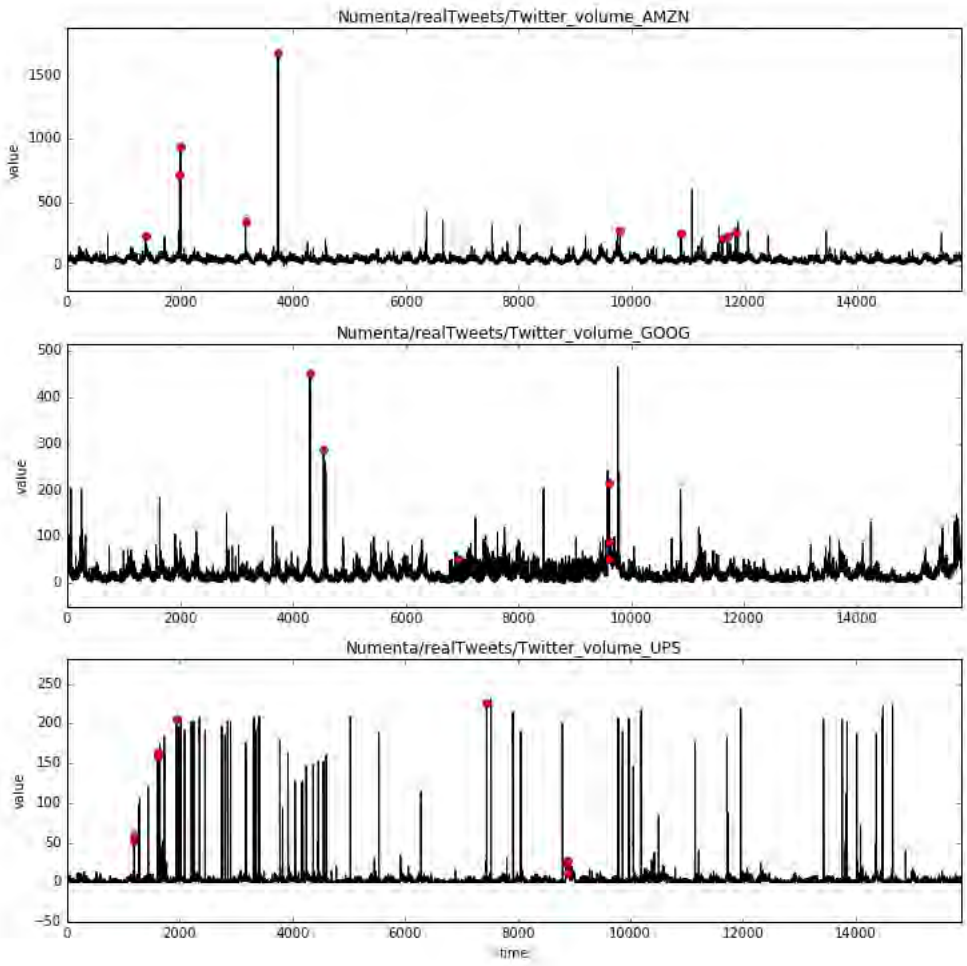


Figure 4.18**:** Three examples of time series data in the collection realTweets from the Numenta benchmark.

The experimental results for the performance of each time series data in the collection realTweets are shown as the Figure 4.19 on the next page.

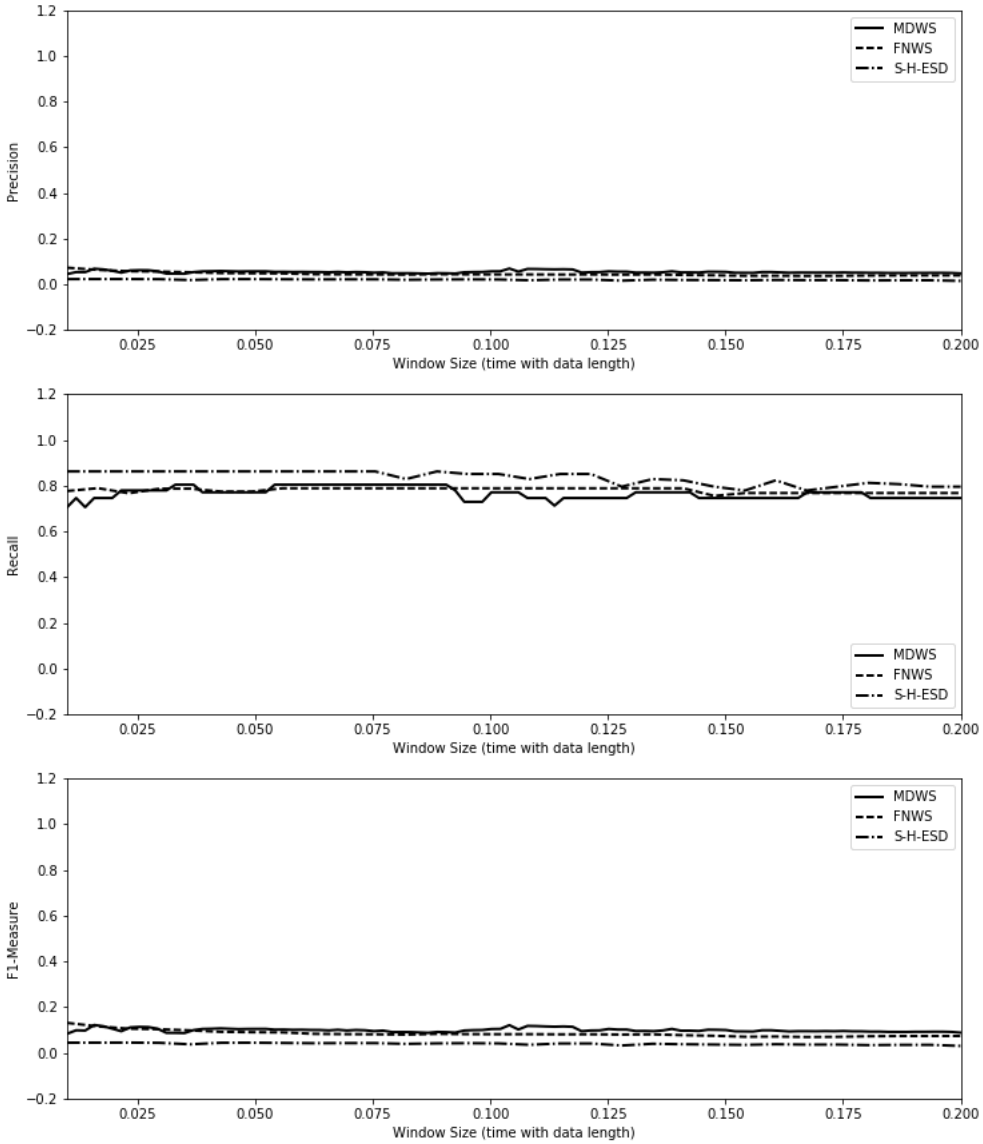


Figure 4.19**:** The performance of MDWS, FNWS and S-H-ESD to detect anomalies on the real world time series data in the collection realTweets from the Numenta benchmark.

4.2 Computing Efficiency

To compare the wall-clock time for computing the anomaly score, the large size of synthetic time series are generated. The experiments will perform on two groups of time series data. The first group varies the length of the time series data while fixed the window length. The second group varies the window length while fixed the length of the time series data.

Note that, in this section only FNWS algorithm is compared with MDWS algorithm, because they were implemented via Python programming language, but S-H-ESD is the R package.

The examples of synthetic time series data are shown in Figure 4.20. They are based on the collection A2Benchmark of the Yahoo! benchmark with trend, seasonality and noise added.

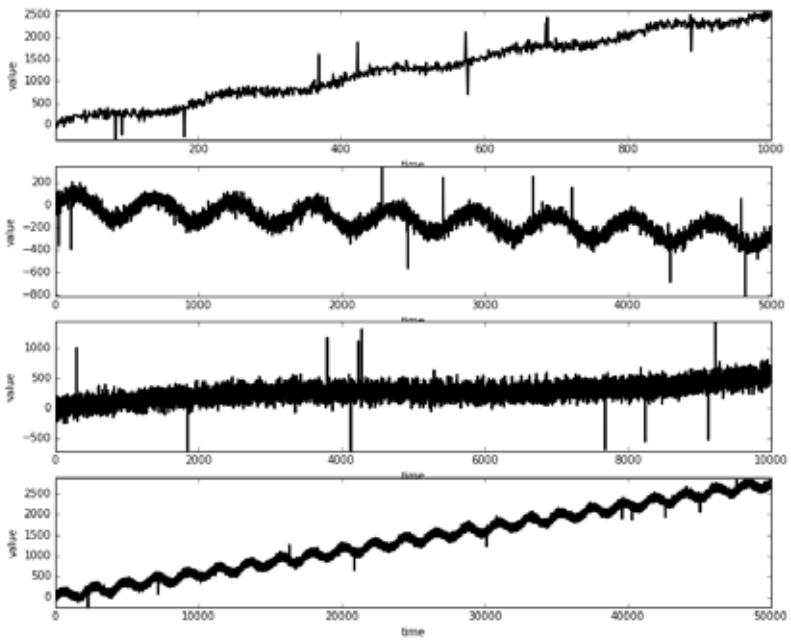


Figure 4.20**:** Four examples of synthetic time series data which are used for testing the computing efficiency of each algorithm.

Vary Data Length

For the first experiment, the window length is fixed to 50 and varies the data length from 1000 to 100000. The result is shown in Figure 4.21. The MDWS algorithm (solid line) shows very small running time for all lengths of the time series data, but it is not true for the FNWS algorithm (dashed line).

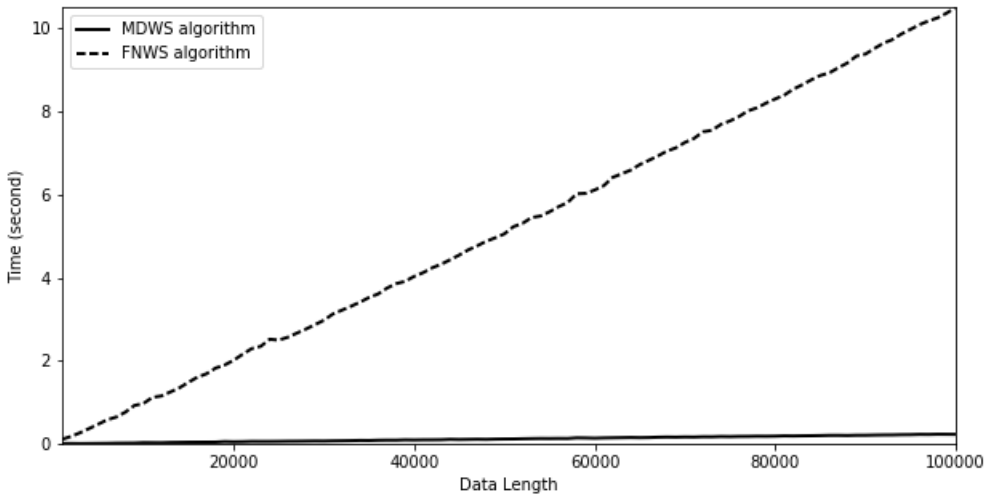


Figure 4.21**:** Running time of two algorithms, i.e. MDWS algorithm (solid line) and FNWS algorithm (dash line) with varies the data length.

Vary Window Length

For the next experiment, the data length is fixed to 10000 and varies the window length from 10 to 1000. The result is shown in Figure 4.22, the MDWS algorithm (solid line) shows very small running time for all lengths of the window subseries, but it is not true for the FNWS algorithm (dashed line).

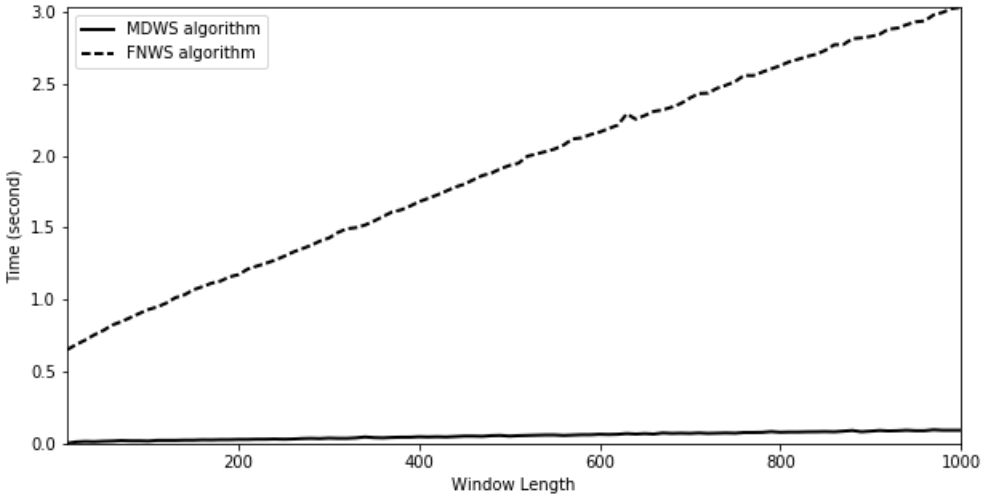


Figure 4.22**:** Running time of two algorithms, i.e. MDWS algorithm (solid line) and FNWS algorithm (dash line) with varies the window length.

Hence, the MDWS algorithm is more efficient than FNWS algorithm with both the large data length and the large window length.

CHAPTER V

CONCLUSION

Median-Difference Window subseries Score or MDWS is proposed in this thesis. It is a novel anomaly score for assigning to a data point in the time series data. The main purpose is to distinguish the difference between the normal data points and the contextual anomalies.

MDWS relies on the fact that, the contextual anomaly on the time series data is distinct from its normal surrounding context along the time dimension. The sliding window is used to determine the context, both preceding and succeeding of each data point which is called the middle-window point. Then, the anomaly score is defined by the difference of the middle-window point and its window subseries. That difference is computed using the subtraction of the middle-window point with the representative of that window subseries which is robust against the influence of anomalies, i.e. the median. Then, MDWS of the normal data points are close to zero, but for the anomalies are very difference from zero.

The analysis found that the MDWS is independent of the trend component. Moreover, the window length should be set smaller than the period length, e.g. the quarter of the period length. To specify a data point to be an anomaly, the thresholds are suggested which are developed from the interquartile range rule. The upper threshold is computed from the set of data points which have large MDWS than the ninth decile of all MDWSs. Then, the interquartile range rule is applied in that set, but only the right tail is evaluated. Similarly, the lower threshold is computed from the set of data points which have small MDWS than the first decile of all MDWSs. Then, the interquartile range rule is applied in that set, but only the left tail is evaluated. Finally, the MDWS algorithm is proposed. It computes the median only in the first window subseries and incrementally

updates the median by the recent data point of the next window which takes only $O(n)$ time complexity, where n is the data length.

For the experiment, MDWS achieves the best accuracy performance on the various synthetic and real world datasets which are compared with other effective methods, i.e. Seasonal Hybrid ESD (S-H-ESD) and Furthest Neighbor Window Subserires (FNWS). Moreover, the MDWS algorithm uses a very small running time for the large time series data and large window length comparing with the FNWS algorithm.

Future Work

Although MDWS provides the best accuracy performance, it still has some limitations which will need to be solved for the future research.

1. MDWS cannot assign scores for the data points at the beginning and the end of the time series data, e.g. Figure 4.8 (bottom).
2. MDWS cannot detect the collective anomalies, for example in Figure 4.8 (middle).
3. MDWS can not handle the multi-distribution time series data, which is shown in the collection A4Benchmark of the Yahoo! benchmark, see Figure 4.6.

Moreover, the MDWS algorithm should be modified to be able to work on the streaming data and Big Data.

REFERENCES

[1] Bollerslev, T. "Generalized autoregressive conditional heteroskedasticity". *Journal of econometrics* 31.3 (1986): 307-327.

[2] Box, G.E., and Jenkins, G.M. *Time series analysis: forecasting and control.* revised ed. Holden-Day, 1976.

[3] Breunig, M.M., et al. "LOF: identifying density-based local outliers". *ACM sigmod record* 29.2 (2000): 93-104.

[4] Brown, R.G. "Exponential smoothing for predicting demand." *Operations Research* 5.1 (1957): 145.

[5] Chandola, V., Cheboli D., and Kumar, V. "Detecting anomalies in a time series database". *Computer Science Department, University of Minnesota* Tech. Rep (2009).

[6] Chandola, V., Banerjee A., and Kumar, V. "Anomaly detection: A survey". *ACM computing surveys (CSUR)* 41.3 (2009): 15:1-15:58.

[7] Cheboli, D. "Anomaly detection of time series.". Diss. University of Minnesota, (2010).

[8] Chuah, M., and Fen F. "ECG anomaly detection via time series analysis". *International Symposium on Parallel and Distributed Processing and Applications* Springer Berlin Heidelberg (2007): 123-135.

[9] Cleveland, R.B., William S.C., and Terpenning, I. "STL: A seasonal-trend decomposition procedure based on loess". *Journal of Official Statistics* 6.1 (1990): 3-73.

[10] Debar, Herve, Monique Becker, and Didier Siboni. "A neural network component for an intrusion detection system". *Research in Security and Privacy, 1992. Proceedings* IEEE Computer Society Symposium on. IEEE, (1992): 240-250.

[11] Ferdousi, Z., and Akira M. "Unsupervised outlier detection in time series data". *Data Engineering Workshops, 2006. Proceedings* 22nd International Conference on. IEEE (2006): 51-56.

[12] Goldstein, M., and Andreas D. "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm". *KI-2012: Poster and Demo Track* (2012): 59-63.

[13] Hampel, F.R. "Contributions to the theory of robust estimation". *Diss. University of California* 1968.

[14] Han, J., Pei, J. and Kamber, M. *Data mining: concepts and techniques*. Elsevier, (2011).

[15] Hawkins, D.M. "Identification of outliers". Vol. 11. London: Chapman and Hall (1980).

[16] Jin, W., et al. "Ranking outliers using symmetric neighborhood relationship". *Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining* Springer Berlin Heidelberg, (2006): 577-593.

[17] Kejariwal, A. "Introducing practical and robust anomaly detection in a time series". Twitter Engineering Blog. Web 15 (2015).

[18] Kitimoon S., and Sinapiromsaran K. "Anomaly Detection on Time Series from Furthest Neighbor Window Subseries". *Proc. International Conference on Applied Statistics 2016* Phuket, Thailand (2016): 199-203.

[19] Knox, E.M., and Ng R.T. "Algorithms for mining distancebased outliers in large datasets". *Proceedings of the International Conference on Very Large Data Bases* (1998): 392-403.

[20] Lavin, A., and Ahmad S. "Evaluating Real-Time Anomaly Detection Algorithms– The Numenta Anomaly Benchmark". *Proc. Machine Learning and Applications (ICMLA) 2015* IEEE 14th International Conference on IEEE (2015): 38-44.

[21] Lin J., et al. “Approximations to magic: Finding unusual medical time series”. *Proc. 18th Computer-Based Medical Systems 2005* IEEE Symposium on. IEEE (2005): 329-334.

[22] Ma, J., and Perkins, S.. “Online novelty detection on temporal sequences”. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* ACM (2003): 613-618.

[23] Michael, C.C., and Ghosh, A. “Two state-based approaches to program-based anomaly detection”. *Computer Security Applications. 2000* ACSAC’00. 16th Annual Conference. IEEE (2000): 21-30.

[24] Protopapas, P., et al. “Finding outlier light curves in catalogues of periodic variable stars”. *Monthly Notices of the Royal Astronomical Society* 369.2 (2006): 677-696.

[25] Ramaswamy, S., Rastogi R., and Shim K. “Efficient algorithms for mining outliers from large data sets”. *ACM Sigmod Record* 29.2 (2000): 427-438.

[26] Rebbapragada, U., et al. “Finding anomalous periodic time series”. *Machine learning* 74.3 (2009): 281-313.

[27] Rosner, B. “On the detection of many outliers”. *Technometrics* 17.2 (1975): 221-227.

[28] Rosner, B. “Percentage points for a generalized ESD many-outlier procedure”. *Technometrics* 25.2 (1983): 165-172.

[29] Vallis, O, Hochenbaum J., and Kejariwal A. “A Novel Technique for Long-Term Anomaly Detection in the Cloud”. *HotCloud* (2014).

[30] Viinikka, J., et al. “Processing intrusion detection alert aggregates with time series modeling”. *Information Fusion* 10.4 (2009): 312-324.

[31] Wei, W.W.S. *Time series analysis.* Reading: Addison-Wesley publ (1994).

[32] Yahoo!. “A Labeled Anomaly Detection Dataset”. version 1.0 [Internet]. Available from: https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70/

[33] Zhang, J., et al. “Detection of outbreaks from time series data using wavelet transform”. *AMIA Annual Symposium Proceedings* American Medical Informatics Association (2003): 748.

BIOGRAPHY

Name Mr. Artit Sagoolmuang

Date of Birth 10 January 1993

Place of Birth Samutsongkhram, Thailand

Education B.S. (Mathematics) (First Class Honours),
Kasetsart University, 2014

Scolarship Science Achievement Scholarship of Thailand (SAST)

Publication

- Sagoolmuang A., and Sinapiromsaran K. "Median-Difference Window Subseries Score for Contextual Anomaly on Time Series". *Proc. The 8th annual International Conference on Information and Communication Technology for Embedded Systems (IC-ICTES 2017)* Chonburi, Thailand (2017): 92-97.